

IMS/ESA



Utilities Reference: Transaction Manager

Version 6

IMS/ESA



Utilities Reference: Transaction Manager

Version 6

Note

Before using this information and the product it supports, be sure to read the general information under "Notices" on page v.

Second Edition (October 1999)

This edition replaces and makes obsolete the previous edition, SC26-8771-00. The technical changes for this edition are summarized under "Summary of Changes" on page xv and are indicated by a vertical bar to the left of a change.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

A form for readers' comments is provided at the back of this publication. If the form has been removed, address your comments to:

IBM Corporation, BWE/H3
P.O. Box 49023
San Jose, CA, 95161-9023
U.S.A.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1974, 1999. All rights reserved.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	v
Trademarks	v
Preface	vii
Summary of Contents	vii
Prerequisite Knowledge	vii
Syntax Diagrams	vii
Supported Environments for Various Utilities	x
Summary of Changes	xv
Changes to the Current Edition of This Book for V6	xv
Changes to This Book for V6.	xv
Library Changes for Version 6	xv

Part 1. Generation Utilities 1

Chapter 1. MFS Language Utility (DFSUPAA0)	3
Standard Mode	4
MFSUTL Procedure Description	5
Batch Mode	9
MFSBTCH1 Procedure Description	10
MFSBTCH2 Procedure Description	11
Test Mode	12
MFSTEST Procedure Description	13
MFS Library Backup and Restore Operations.	15
MFSBACK Procedure	15
MFSREST Procedure	17
JCL Parameter Descriptions	19
DDNAMES Used in MFS Procedures	21
MFSUTL and MFSTEST Region Parameter Estimate.	22
Chapter 2. MFS Device Characteristics Table Utility (DFSUTB00)	23
Restrictions	23
MFSDCT Procedure	23
PROC Statement	25
EXEC Statement	26
DD Statements	27
MFS Device Descriptors	27
Error Processing	28

Part 2. Service Utilities 29

Chapter 3. MFS Service Utility (DFSUTSA0)	31
Restrictions	32
MFSRVC Procedure	32
PROC Statement	32
EXEC Statement	33
DD Statements	33
Invoking the Procedure	33
Function Descriptions	33
INDEX	34
DELETE	34

SCRATCH	35
RELATE	37
LIST.	37
LIST the MFS Device Characteristics Table	40
LIST DEVCHAR Output.	41
Utility Control Statements	41
Positional Parameters	41
Keywords	42
 Chapter 4. Multiple Systems Verification Utility (DFSUMSV0)	45
Restrictions	45
Prerequisites.	46
IMSMSV Procedure	46
PROC Statement	47
EXEC Statement	47
Invoking the Procedure	47
Input.	48
Input Validation	48
Multisystem Control Block Verification	48
Output Messages and Path Map	50
Utility Control Statements	52
Error Processing	53
 Chapter 5. Spool SYSOUT Print Utility (DFSUPRT0)	55
Restrictions	55
Input and Output	55
DFSWTnnn Procedure	55
PROC Statement	56
EXEC Statement	56
DD statements	56
IMSWTnnn Job.	57
Sample Output	57
Error Processing	57
 Chapter 6. Time-Controlled Operations Verification Utility (DFSTVER0)	59
TCO Verification Procedure	59
EXEC Statement	60
DD Statements	61
Output	61
Error Report	61
Statistics Report	62
Timer-Elements Report	62
Message-Table Report	63
Summary Report	63
Return Codes	64

Part 3. Appendixes.	65
--------------------------------------	-----------

IMS/ESA Version 6 Library	67
Index	69
Readers' Comments — We'd Like to Hear from You	73

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
U.S.A.

Licensees of this program who wish to have information about it for the purpose of enabling (1) the exchange of information between independently created programs and other programs (including this one) and (2) the mutual use of the information that has been exchanged, should contact:

IBM Corporation
555 Bailey Avenue, W92/H3
P.O. Box 49023
San Jose, CA 95161-9023

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Trademarks

The following terms in this publication are trademarks of the IBM Corporation in the United States or other countries or both:

ACF/VTAM	IMS/ESA
CICS	MVS
IBM	MVS/ESA
IMS	RETAIN
IMS Client Server/2	VTAM

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Preface

This book is a reference for the IMS/ESA Version 6 data communications utilities. This book tells the IMS systems programmer how to use the utilities to define and maintain data communications.

Summary of Contents

Part 1. Generation Utilities, describes the utilities used during the system generation process:

- The MFS Language utility is used to create and store the message format service control blocks.
- The MFS Device Characteristics Table utility is used to define new screen sizes in a descriptor member of the IMS.PROCLIB library without performing an IMS system generation.

Part 2. Service Utilities, describes the service utilities:

- The MFS Service utility is used to control and maintain MFS intermediate control blocks after they are processed and stored by the MFS Language utility.
- The Multiple Systems Verification utility is used to verify the consistency and compatibility of system definitions for IMS systems in a multisystem environment. Use this utility with IMS multiple systems coupling when MTM, CTC, or VTAM is used.
- The Spool SYSOUT Print utility is used to copy messages produced by the online control program to a system output device when a communication line is defined for Spool SYSOUT during system definition.
- The Time-Controlled Operations Verification utility is used to ensure error-free TCO script members.

A complete list of the IMS publications is found on the back cover of this book.

Prerequisite Knowledge

IBM offers a wide variety of classroom and self-study courses to help you learn IMS. For a complete list of courses, see the IMS home page on the World Wide Web at: <http://www.software.ibm.com/data/ims>

The reader must understand IMS concepts and facilities, data communication, and the access methods used by IMS and MVS.

Syntax Diagrams

The following rules apply to the syntax diagrams used in this book:

Arrow symbols

Read the syntax diagrams from left to right, from top to bottom, following the path of the line.

- ▶— Indicates the beginning of a statement.
- ▶ Indicates that the statement syntax is continued on the next line.
- ▶— Indicates that a statement is continued from the previous line.
- ▶◀ Indicates the end of a statement.

Diagrams of syntactical units other than complete statements start with the \blacktriangleright symbol and end with the \blacktriangleright symbol.

Conventions

- Keywords, their allowable synonyms, and reserved parameters, appear in uppercase for MVS and OS/2 operating systems, and lowercase for UNIX operating systems. These items must be entered exactly as shown.
- Variables appear in lowercase italics (for example, *column-name*). They represent user-defined parameters or suboptions.
- When entering commands, separate parameters and keywords by at least one blank if there is no intervening punctuation.
- Enter punctuation marks (slashes, commas, periods, parentheses, quotation marks, equal signs) and numbers exactly as given.
- Footnotes are shown by a number in parentheses, for example, (1).
- A $\text{\texttt{b}}$ symbol indicates one blank position.

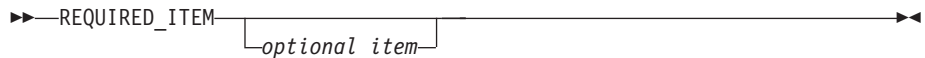
Required items

Required items appear on the horizontal line (the main path).



Optional Items

Optional items appear below the main path.

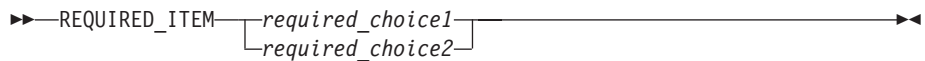


If an optional item appears above the main path, that item has no effect on the execution of the statement and is used only for readability.

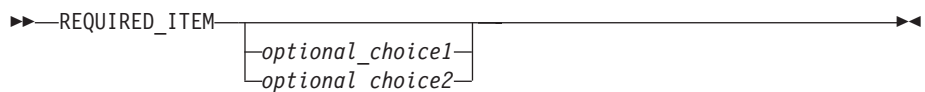


Multiple required or optional items

If you can choose from two or more items, they appear vertically in a stack. If you *must* choose one of the items, one item of the stack appears on the main path.



If choosing one of the items is optional, the entire stack appears below the main path.

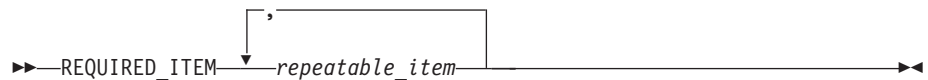


Repeatable items

An arrow returning to the left above the main line indicates that an item can be repeated.



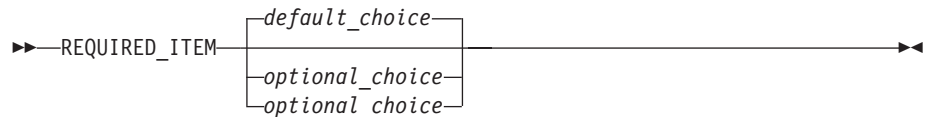
If the repeat arrow contains a comma, you must separate repeated items with a comma.



A repeat arrow above a stack indicates that you can specify more than one of the choices in the stack.

Default keywords

IBM-supplied default keywords appear above the main path, and the remaining choices are shown below the main path. In the parameter list following the syntax diagram, the default choices are underlined.



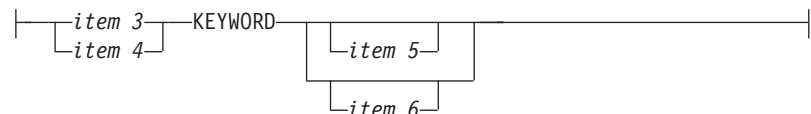
IMS-specific syntax information

Fragments

Sometimes a diagram must be split into fragments. The fragments are represented by a letter or fragment name, set off like this: | A |. The fragment follows the end of the main diagram. The following example shows the use of a fragment.



A:

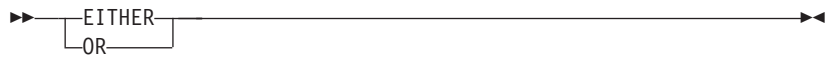


Substitution-block

Sometimes a set of several parameters is represented by a substitution-block such as <A>. For example, in the imaginary /VERB command you could enter /VERB LINE 1, /VERB EITHER LINE 1, or /VERB OR LINE 1.

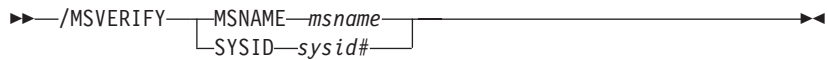


where <A> is:



Parameter endings

Parameters with number values end with the symbol '#', parameters that are names end with 'name', and parameters that can be generic end with '*'.



The MSNAME keyword in the example supports a name value and the SYSID keyword supports a number value.

Supported Environments for Various Utilities

Table 1, Table 2 on page xi, and Table 3 on page xii provide a comprehensive listing of all the IMS/ESA utilities contained in all three of the Utilities Reference books. The figures also indicate whether the utility supports either DBCTL or DCCTL or both.

Table 1 lists the database utilities described in the *Utilities Reference: Database Manager*.

Table 1. Listing of Database Utilities and Supported Environments

Utility or Report Name	Module Name	Supports DBCTL	Supports DCCTL
Reorganization Utilities:			
HISAM Reorganization Unload utility	DFSURUL0	X	
HISAM Reorganization Reload utility	DFSURRL0	X	
HD Reorganization Unload utility	DFSURGU0	X	
HD Reorganization Reload utility	DFSURGL0	X	
Database Surveyor utility	DFSPRSUR	X	
Partial Database Reorganization utility	DFSPRCT1 and DFSPRCT2	X	
Database Preorganization utility	DFSURPR0	X	
Database Scan utility	DFSURGS0	X	
Database Prefix Resolution utility	DFSURG10	X	
Database Prefix Update utility	DFSURGP0	X	

Table 1. Listing of Database Utilities and Supported Environments (continued)

Utility or Report Name	Module Name	Supports DBCTL	Supports DCCTL
MSDB Maintenance utility	DBFDBMA0		
DEDB Initialization utility	DBFUMIN0	X	
DEDB Sequential Dependent Scan utility	DBFUMSC0	X	
DEDB Sequential Dependent Delete utility	DBFUMDL0	X	
High Speed DEDB Direct Reorganization utility	DBFUHDR0	X	
Backup Utilities:			
Database Image Copy utility	DFSUDMP0	X	
Online Database Image Copy utility	DFSUICP0	X	
Recovery Utilities:			
Database Change Accumulation utility	DFSUCUM0	X	
Database Recover utility	DFSURDB0	X	
Batch Backout utility	DFSBB000	X	
MSDB Dump Recovery utility	DBFDBDR0		
DEDB Area Data Set Create utility	DBFUMRI0	X	
DEDB Area Data Set Compare utility	DBFUMMH0		X
Conversion Utilities:			
MSDB-to-DEDB Conversion utility	DBFUCDB0		
Utility Control:			
Utility Control Facility	DFSUCF00	X	
Report and Test Utilities:			
Program-Isolation-Trace Report utility	DFSPIRP0	X	
Database-Monitor Report Print utility	DFSUTR30		
Sequential Buffer Test utility	DFSSBHD0		
Report Interpretation:			
Interpreting Database Monitor Reports			

Table 2 lists the data communications utilities described in the *Utilities Reference: Transaction Manager*.

Table 2. Listing of Data Communications Utilities and Supported Environments

Utility or Report Name	Module Name	Supports DBCTL	Supports DCCTL
Generation Utilities:			
MFS Language utility	DFSUPAA0		X
Reorganization Utilities:			

Table 2. Listing of Data Communications Utilities and Supported Environments (continued)

Utility or Report Name	Module Name	Supports DBCTL	Supports DCCTL
MFS Device Characteristics Table utility	DFSUTB00		X
Service Utilities:			
Spool SYSOUT Print utility	DFSUPRT0		X
Multiple Systems Verification utility	DFSUMSV0		X
MFS Service utility	DFSUTSA0		X
Time-Controlled Operations Verification utility	DFSTVER0		X

Table 3 lists the system utilities described in the *Utilities Reference: System*.

Table 3. Listing of System Utilities and Supported Environments

Utility or Report Name	Module Name	Supports DBCTL	Supports DCCTL
Generation Utilities:			
Database Description (DBD) Generation utility	DBDGEN	X	X
Program Specification Block (PSB) Generation utility	PSBGEN	X	X
Application Control Block (ACB) Maintenance utility	ACBGEN	X	X
Service Utilities:			
Dynamic Allocation Macro utility	DFSMDA	X	X
Online Change utility	DFSUOCU0	X	X
Security Maintenance utility	DFSISMP0	X	X
Dynamic SVC utility	DFSUSVC0	X	X
Log Utilities:			
Log Recovery utility	DFSULTR0	X	X
Log Archive utility	DFSUARC0	X	X
Log Merge utility	DFSMTMG0		X
Analysis and Report Utilities:			
File Select and Formatting Print utility	DFSERA10	X	X
Offline Dump Formatter utility	DFSOFMD0	X	X
Statistical Analysis utility	DFSISTS0		X
Log Transaction Analysis utility	DFSILTA0		X
Fast Path Log Analysis utility	DBFULTA0	X	X
IMS-Monitor Report Print utility	DFSUR20	X	X
Report Interpretation:			
Interpreting Statistical Analysis and Log Transaction Reports			X

Table 3. Listing of System Utilities and Supported Environments (continued)

Utility or Report Name	Module Name	Supports DBCTL	Supports DCCTL
Interpreting //DFSSTAT Reports		X	X
Interpreting IMS Monitor Reports			
Interpreting IMS Monitor Reports for DBCTL		X	
Interpreting IMS Monitor Reports for DCCTL			X

Summary of Changes

Changes to the Current Edition of This Book for V6

This second edition includes new technical information for Version 6, as well as technical and editorial changes to the first edition.

Changes to This Book for V6

Minor editorial changes have been made to this book.

Library Changes for Version 6

The IMS/ESA Version 6 library differs from the IMS/ESA Version 5 library in these major respects:

- *IMS/ESA Common Queue Server Guide and Reference*
This new book describes the IMS Common Queue Server (CQS).
- *IMS/ESA DBRC Guide and Reference*
This new book describes all the functions of IMS Database Recovery Control (DBRC).
- The IMS Application Programming summary books (*IMS/ESA Application Programming: Database Manager Summary*, *IMS/ESA Application Programming: Transaction Manager Summary*, and *IMS/ESA Application Programming: EXEC DLI Commands for CICS and IMS Summary*) are no longer included with the IMS library.
- The Softcopy Master Index is not included.
- All information about IRLM 1.5 and data sharing using IRLM 1.5 has been removed from the IMS V6 books. If you use IRLM 1.5, and want to migrate to using IRLM 2.1 and Sysplex data sharing, see *IMS/ESA Release Planning Guide*.
- The chapter that was titled "Database Control (DBCTL) Interface" in the *IMS/ESA Customization Guide* has been revised for Open Database Access (ODBA) and moved to "Appendix A, Using the Database Resource Adapter (DRA)" in the *IMS/ESA Application Programming: Database Manager*.

Part 1. Generation Utilities

Chapter 1. MFS Language Utility (DFSUPAA0)	3
Standard Mode	4
MFSUTL Procedure Description	5
Step 1 (S1)	5
Step 2 (S2)	7
JCL Requirements	8
Batch Mode	9
MFSBTCH1 Procedure Description	10
JCL Requirements	10
MFSBTCH2 Procedure Description	11
JCL Requirements	11
Test Mode	12
MFSTEST Procedure Description	13
Step 1 (S1)	13
Step 2 (S2)	14
JCL Requirements	14
MFS Library Backup and Restore Operations	15
MFSBACK Procedure	15
MFSREST Procedure	17
JCL Parameter Descriptions	19
DDNAMES Used in MFS Procedures	21
MFSUTL and MFSTEST Region Parameter Estimate	22
 Chapter 2. MFS Device Characteristics Table Utility (DFSUTB00)	 23
Restrictions	23
MFSDCT Procedure	23
PROC Statement	25
EXEC Statement	26
DD Statements	27
MFS Device Descriptors	27
MFS Device Descriptor Format	28
Error Processing	28

Chapter 1. MFS Language Utility (DFSUPAA0)

This chapter explains how to use the MFS Language utility to create and store the message format service (MFS) control blocks. The intermediate text block (ITB) form of the control blocks is placed in the IMS.REFERAL library. The control blocks are placed in the IMS.FORMAT library for use during normal IMS operation.

Definition: One format and all the messages that refer to it in their SOR= operand make up a *format set*.

The MFS Language utility has three modes of operation: standard, test, and batch. In all three modes, the utility is executed offline, accepts the same control statements, and produces the same kinds of ITBs and control blocks. The modes differ in their use of the MFS libraries. Accordingly, they use different procedures. The JCL for all the MFS utility procedures is shown later in this chapter.

In standard mode, ITBs written in IMS.REFERAL are converted to control blocks and placed in the staging library, IMS.FORMAT, by the MFSUTL procedure. Because the control blocks are placed in the staging library and not the active library, the standard mode can run concurrently with the online IMS control region.

Batch mode differs from standard mode, in that the MFSBTCH1 procedure places the created control blocks in a special library, IMS.MFSBTCH, for later transfer by the MFSBTCH2 procedure (in another job) to the staging library, IMS.FORMAT.

In test mode, the MFSTEST procedure creates control blocks and places them in a separate IMS.TFORMAT library. The control blocks can be tested without interfering with online operation and can operate concurrently with the online IMS control region.

The MFSTEST procedure should not be executed concurrently with itself or any other program or procedure that utilizes the MFS libraries. To test the control blocks in IMS.TFORMAT, the terminal operator enters the /TEST MFS command. Then, test control blocks from IMS.TFORMAT (as well as online control blocks from the active format library, if necessary) are read into a buffer for test operation. After successful testing, the control blocks can be placed in the staging IMS.FORMAT library by recompiling the source statements using the MFSUTL procedure.

Stage 2 of IMS system definition generates the following procedures:

MFSUTL	A two-step standard mode execution procedure of the MFS Language utility for creating MFS online control blocks and placing these blocks into the IMS.FORMAT library.
MFSBTCH1	A one-step batch mode execution procedure of the MFS Language utility for creating and accumulating MFS online blocks.
MFSBTCH2	A one-step batch mode execution procedure of the MFS Language utility for placing the accumulated MFS online control blocks (from MFSBTCH1) into the IMS.FORMAT library.
MFSBACK	A two-step execution procedure to back up the MFS libraries. If the optional MFSTEST facility is used, MFSBACK contains an additional step to back up the test library.
MFSREST	A two-step execution procedure to restore the MFS libraries. If the

MFS Language Utility

optional MFSTEST facility is used, MFSREST contains an additional step to restore the test library.

MFSRVC A one-step execution procedure for maintaining the MFS libraries.

If MFSTEST mode is selected during system definition, an additional procedure is generated:

MFSTEST A two-step test mode execution procedure of the MFS Language utility for creating MFS online blocks and placing them into the IMS.TFORMAT library.

In addition to the procedures for creating new or replacement control blocks, the MFS Language utility includes MFSBACK and MFSREST procedures for backup and restore operations in MFS libraries.

Delete and listing operations are performed by the service utility.

Related Reading:

- MFS control blocks are described in *IMS/ESA Application Programming: Design Guide* .
- The control statements used by the MFS Language utility are described in *IMS/ESA Application Programming: Transaction Manager* .
- The MFSRVC procedure is described in “Chapter 3. MFS Service Utility (DFSUTSA0)” on page 31. The rest of the procedures listed above are described in this chapter.
- The Delete and listing operations performed by the service utility, are described in “Chapter 3. MFS Service Utility (DFSUTSA0)” on page 31.

In this Chapter:

- “Standard Mode”
- “Batch Mode” on page 9
- “Test Mode” on page 12
- “MFS Library Backup and Restore Operations” on page 15
- “JCL Parameter Descriptions” on page 19
- “MFSUTL and MFSTEST Region Parameter Estimate” on page 22

Standard Mode

This section describes the two-step mode of operation using the MFSUTL procedure.

Figure 1 on page 5 shows an overview of the two-step standard mode of operation using the MFSUTL procedure. For the JCL for the MFSUTL procedure see “JCL Requirements” on page 8.

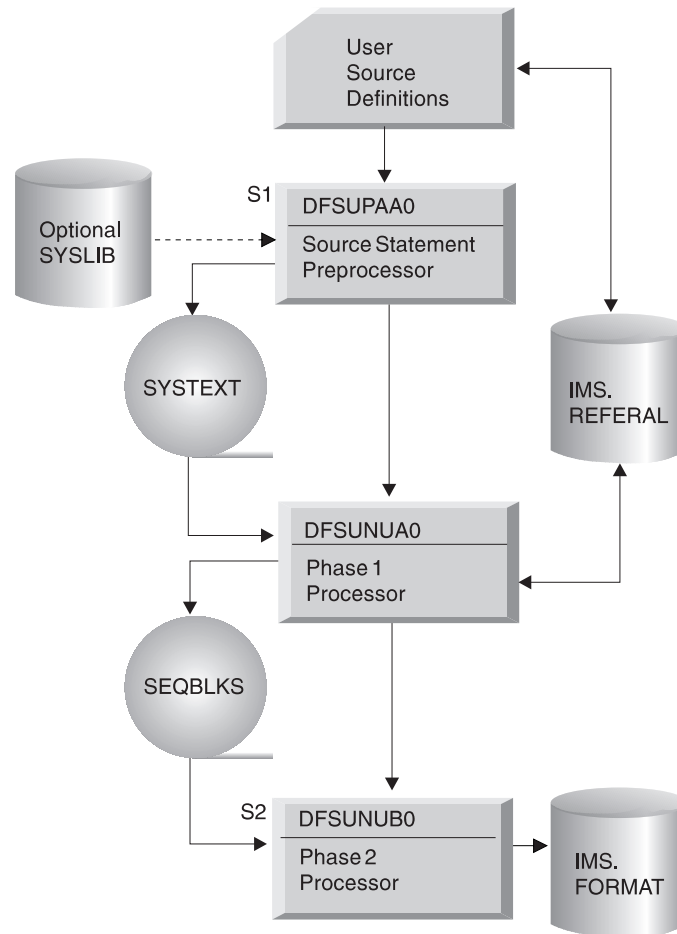


Figure 1. Overall Flow with the MFSUTL Procedure

The following sections discuss the details of Step 1 (S1) and Step 2 (S2).

MFSUTL Procedure Description

Step 1 (S1)

Source Statement Preprocessor: The MFS Language utility preprocessor provides a nonassembler, nonmacro preprocessor for the MFS Language source statements.

Related Reading: See information on utility control statements in *IMS/ESA Application Programming: Design Guide*.

The execution of the preprocessor generates intermediate text blocks (ITBs), which are then processed by the remaining utility phase to generate message (MSG) and format (FMT) control blocks. IMS uses the control blocks to format messages for device display and application program presentation.

The primary function of the preprocessor is to perform syntax and relational validity checks on user specifications. The preprocessor generates ITBs for each MSG, FMT, partition descriptor block (PDB), and TABLE source definition processed, and stores them in IMS.REFERAL.

MFS Language Utility

The IMS.REFERAL partitioned data set (PDS) directory contains an entry for each MSG, FMT, PDB, and TABLE ITB. Additionally, the interrelationships between all known FMT and MSG ITBs that have been placed in the IMS.REFERAL are recorded in the PDS directory.

The preprocessor executes in the following order:

1. The preprocessor constructs a control table representing the interrelationships between all known FMT and MSG ITBs that have been placed in IMS.REFERAL. If you request the compress function in the EXEC statement, the preprocessor compresses IMS.REFERAL.
2. The preprocessor adds the user-supplied FMT and MSG definitions to the control table and resolves them against the table to ensure that, when a given definition is supplied for processing, all control blocks in the format set are reprocessed by phase 1. This resolution also allows you to compile source for only the message or format that requires change—not the entire format set.
3. After the resolution function has been accomplished to determine the definitions to be processed, the preprocessor places the control statements for these FMT and MSG definitions on the SYSTEXT data set for phase 1 processing.

If you change a PDB definition, each format set referencing that PDB might need to be recompiled.

The following actions cause phase 1 reprocessing of the format set ITBs to create new FMT and MSG control blocks for the IMS.FORMAT library:

- Modification of a FMT definition that already exists as a control block in IMS.FORMAT.
- Modification of a MSG definition that already exists as a control block in IMS.FORMAT.
- Addition of a new MSG to the format set.
- Reassignment of a MSG definition to a different FMT. This reassignment causes the old format set and the new format set to be reprocessed.

If a MSG definition refers to a FMT name (through the SOR= keyword) that has not yet been supplied to the MFS Language utility, the preprocessor stores the MSG ITB in the IMS.REFERAL library. The MSG control block is not created until a FMT definition is supplied. The format set is then processed to create the new MSG and FMT control blocks for IMS.FORMAT.

Similarly, if a FMT definition is supplied to the MFS Language utility and no MSGs refer to it, the FMT ITB is stored in IMS.REFERAL. The FMT control block is not created until at least one MSG definition is supplied. The format set is then processed to create the new MSG and FMT control blocks for IMS.FORMAT.

Each IMS error message is accompanied by a return code of 4, 8, 12, 16, or 20 to indicate increasing severity of error.

Related Reading: Refer to *IMS/ESA Messages and Codes* for more information.

If an error condition is detected during the processing of statements that would create an FMT, MSG, PDB, or TABLE ITB, the highest such severity code associated with the message stating the error is kept and used to determine if the ITB is to be written to the IMS.REFERAL library. The preprocessor maintains the highest return code issued for each definition processed. You can specify a compare value (the lowest unacceptable return code) in the STOPRC parameter of

the EXEC statement. If the return code is greater than or equal to the STOPRC value, the ITB is not written in IMS.REFERAL. For example, a STOPRC of 4 permits only ITBs that have a return code of 0 to be written. If no STOPRC is specified, a value of 8 is assumed, and only ITBs having a return code of 0 or 4 are written.

The preprocessor also maintains the highest return code for all ITBs processed during a job. Phase 1 is not given control by the preprocessor if the highest return code is greater than or equal to 16, or if no ITBs were written in IMS.REFERAL. If the return code is 16 or greater, the preprocessor returns control to MVS with a completion code equal to the return code.

Phase 1 Processor: The preprocessor invokes the phase 1 processor. Initially, the phase 1 processor uses the control statements placed by the preprocessor on the SYSTEXT data set to construct a module table representing all of the FMT and MSG ITBs to be processed in this run. After constructing the module table, the phase 1 processor reads in ITBs from IMS.REFERAL and builds control blocks for each MSG and FMT definition. If a TABLE of control functions is requested by an input format definition, the phase 1 processor obtains the TABLE ITB from IMS.REFERAL and builds functions into the device input format (DIF).

When a format definition requests a partition set, the phase 1 processor gets the PDB ITB from IMS.REFERAL and builds the partitioning control functions into the device output format (DOF).

The phase 1 processor places the newly constructed control blocks on the SEQBLKS data set. Each member processed has a control record placed on the SEQBLKS data set identifying the member, its size, and the date and time of creation. This control record is followed by the image of the control block as constructed by the phase 1 processor.

If an error is detected during control block building, an error control record is placed on the SEQBLKS data set for the definition in error, identifying the member in error, and the date and time the error control record was created. In addition, the phase 1 processor returns a completion code of 12 to MVS. If execution of step 2 is forced, the phase 2 processor deletes control blocks with build errors.

The phase 1 processor maintains a high return code for all ITBs processed during an execution of the MFS Language utility. Before returning to MVS, the phase 1 processor compares its high return code to the preprocessor's high return code. The highest of the two is passed to MVS as the completion code for step 1.

Step 2 (S2)

Phase 2 Processor: The phase 2 processor receives control as a job step when the phase 1 processor is finished. The phase 2 processor operates in a two-pass mode to place the new control blocks into the IMS.FORMAT library. On the first pass, the phase 2 processor reads the SEQBLKS data set and creates an internal table that contains the name of every MOD, MID, DOF, and DIF created by the phase 1 processor. The name of the format or message description that had build errors during the phase 1 processor's execution is also added to this internal table. Control blocks in the IMS.FORMAT library that are to be replaced in IMS.FORMAT, or had build errors during phase 1, are deleted from IMS.FORMAT.

If you request the compress function, the phase 2 processor compresses the IMS.FORMAT library. This ensures maximum available library space for adding

MFS Language Utility

control block members and, due to the reprocessing of all related members by the phase 1 processor, allows the grouping of related control blocks for seek time reductions when fetching the control blocks during online execution.

In the second pass, the SEQBLKS data set is reprocessed, together with the module table, to write the new control blocks into IMS.FORMAT and STOW them for a directory update.

If control blocks with entries in the main-storage index directory, \$\$IMSDIR, were deleted and not replaced, the index entries should be deleted. This update can be done automatically, but it is inefficient for a large format library with a relatively small number of blocks deleted. To avoid this, the following parameter can be used for both the MFSUTL and MFSBTCH2 procedures:

DIRUPDT=UPDATE|NOUPDATE. The default is UPDATE. If UPDATE is specified or defaulted, \$\$IMSDIR is updated automatically. If NOUPDATE is specified, updating is bypassed; and you must delete the blocks from \$\$IMSDIR with the MFS Service Utilities.

If index entries are to be added to \$\$IMSDIR for new control blocks created in this run, the INDEX function of the MFS service utility must be used.

The phase 2 processor passes a completion code to MVS for step 2 based on all the control block maintenance to IMS.FORMAT for a given execution of the MFS Language utility.

JCL Requirements

The JCL for the MFSUTL is shown in Figure 2. Refer to “JCL Parameter Descriptions” on page 19 for details on the EXEC parameters and DD statements.

```
//          PROC RGN=360K,SOUT=A,SNODE=IMS,SYS2=,
//          SOR=NOLIB,MBR=NOMBR,PXREF=NOXREF,
//          PCOMP=NOCOMP,PSUBS=NOSUBS,PDIAG=NODIAG,
//          COMPR=NOCOMPRESS,COMPR2=COMPRESS,
//          LN=55,SN=8,DEVCHAR=0,COMPR3=NOCOMPRESS,
//          DIRUPDT=UPDATE
//S1        EXEC PGM=DFSUPAA0,REGION=&RGN,
//          PARM=(&PXREF,&PCOMP,&PSUBS,&PDIAG,&COMPR,
//          'LINECNT=&LN,STOPRC=&SN,DEVCHAR=&DEVCHAR')
//STEPLIB DD DSN=IMS.&SYS2.RESLIB,DISP=SHR
//*SYSLIB - USER OPTION
//SYSIN DD DSN=&SNODE..&SOR.(&MBR),DISP=SHR
//REFIN DD DSN=IMS.&SYS2.REFERAL,DISP=OLD 1
//REFOUT DD DSN=IMS.&SYS2.REFERAL,DISP=OLD 1
//REFRD DD DSN=IMS.&SYS2.REFERAL,DISP=OLD 1
//SYSTEXT DD DSN=&TXTPASS,UNIT=SYSDA,;
//          SPACE=(CYL,(1,1)),DCB=BLKSIZE=800
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//DUMMY DD DSN=IMS.&SYS2.PROCLIB(REFCPY),DISP=SHR
```

Figure 2. MFSUTL Procedure (Part 1 of 2)

```

//UTPRINT DD  SYSOUT=&SOUT
//SYSPRINT DD  SYSOUT=&SOUT,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYSUDUMP DD  SYSOUT=&SOUT
//SEQBLKS DD  DSN=&&BLKS,DISP=(NEW,PASS),
//              UNIT=SYSDA,SPACE=(CYL,(1,1))
//S2      EXEC PGM=DFSUNUB0,REGION=&RGN,
//              PARM=(&COMPR2,&COMPR3,&DIRUPDT,
//              'DEVCHAR=&DEVCHAR'),COND=(8,LT,S1)
//STEPLIB DD  DSN=IMS.&SYS2.RESLIB,DISP=SHR
//SEQBLKS DD  DSN=&&BLKS,DISP=(OLD,DELETE)
//UTPRINT DD  SYSOUT=&SOUT,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYSUDUMP DD  SYSOUT=&SOUT
//FORMAT  DD  DSN=IMS.&SYS2.FORMAT,DISP=SHR
//DUMMY   DD  DSN=IMS.&SYS2.PROCLIB(FMTCPY),DISP=SHR
//SYSPRINT DD  SYSOUT=&SOUT
//SYSUT3  DD  UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4  DD  UNIT=SYSDA,SPACE=(CYL,(1,1))

```

Figure 2. MFSUTL Procedure (Part 2 of 2)

The DISP=OLD specifications are required.

Restriction: A DD DUMMY specification is not supported.

REFCPY Control Statement: The MFSUTL procedure uses this control statement to compress REFERAL.

COPY INDD=REFOUT,OUTDD=REFOUT

FMTCPY Control Statement: The MFSUTL procedure uses this control statement to compress FORMAT.

COPY INDD=FORMAT,OUTDD=FORMAT

Batch Mode

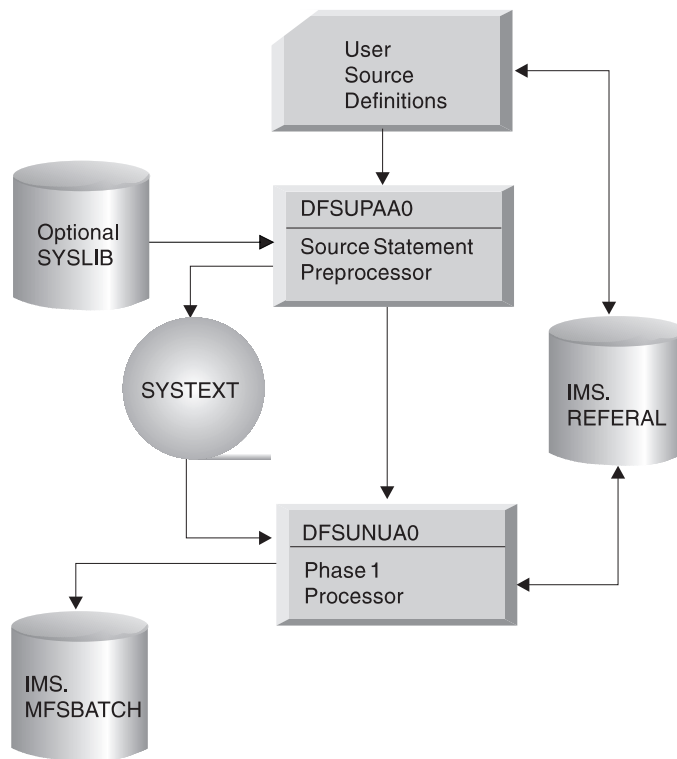
This section describes the two procedures for batch mode: MFSBTCH1 and MFSBTCH2.

Batch mode provides the ability to batch the message descriptors and device formats into an accumulation data set, IMS.MFSBATCH. This data set can then be applied to the MFS staging library, IMS.FORMAT, with a separate job. The batch accumulation data set requires you to allocate and catalog an IMS system data set, IMS.MFSBATCH, large enough to hold all the control blocks that are to be accumulated before they are placed into IMS.FORMAT. See Figure 3 on page 10 for an overview of the batch mode.

Two procedures are required for batch execution: MFSBTCH1 and MFSBTCH2. For the JCL of the MFSBTCH1 procedure, see Figure 4 on page 11. For the JCL of the MFSBTCH2 procedure, see Figure 5 on page 12.

MFS Language Utility

MFSBTCH1



MFSBTCH2

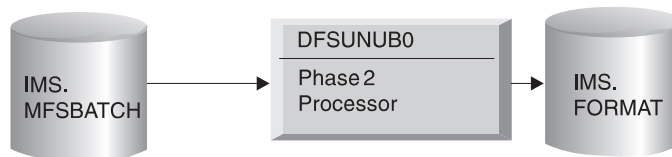


Figure 3. Overall Flow with the MFSBTCH1 and MFSBTCH2 Procedures

MFSBTCH1 Procedure Description

This procedure is identical to step 1 of the MFSUTL procedure on page 5, except that the newly constructed control blocks or error control records, or both, are added to the SEQBLKS accumulation data set, IMS.MFSBATCH.

JCL Requirements

The JCL for the MFSBTCH1 procedure is shown in Figure 4 on page 11. Refer to “JCL Parameter Descriptions” on page 19 for details on the EXEC parameters and DD statements.

```

//          PROC RGN=360K,SOUT=A,SNODE=IMS,SYS2=,
//          SOR=NOLIB,MBR=NOMBR,PXREF=NOXREF,
//          PCOMP=NOCOMP,PSUBS=NOSUBS,PDIAG=NODIAG,
//          COMPR=NOCOMPRESS,LN=55,SN=8,DEVCHAR=0
//S1        EXEC PGM=DFSUPAA0,REGION=&RGN,
//          PARM=(&PXREF,&PCOMP,&PSUBS,&PDIAG,&COMPR,
//          'LINECNT=&LN,STOPRC=&SN,DEVCHAR=&DEVCHAR')
//STEPLIB DD DSN=IMS.&SYS2.RESLIB,DISP=SHR
//*SYSLIB - USER OPTION
//SYSIN DD DSN=&SNODE..&SOR.(&MBR),DISP=SHR
//REFIN DD DSN=IMS.&SYS2.REFERAL,DISP=OLD
//REFOUT DD DSN=IMS.&SYS2.REFERAL,DISP=OLD
//REFRD DD DSN=IMS.&SYS2.REFERAL,DISP=OLD
//SYSTEXT DD DSN=&TXTPASS,UNIT=SYSDA,;
//          SPACE=(CYL,(1,1)),DCB=BLKSIZE=800
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//DUMMY DD DSN=IMS.&SYS2.PROCLIB(REFCPY),DISP=SHR
//UTPRINT DD SYSOUT=&SOUT
//SYSPRINT DD SYSOUT=&SOUT,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYSUDUMP DD SYSOUT=&SOUT
//SEQBLKS DD DSN=IMS.&SYS2.MFSBATCH,DISP=(MOD,KEEP)

```

Figure 4. MFSBTCH1 Procedure

REFCPY Control Statement: The MFSBTCH1 procedure uses this control statement to compress REFERAL.

COPY INDD=REFOUT,OUTDD=REFOUT

MFSBTCH2 Procedure Description

This procedure is identical to step 2 of the MFSUTL procedure on page 7, except as noted in the following paragraphs.

If control blocks with duplicate names are found, only the last one found is recorded. This ensures that if the control block with the same name was processed more than once by the MFSBTCH1 procedure, the control block created last is added to IMS.FORMAT. If the control blocks are to be replaced in IMS.FORMAT, they are first deleted from IMS.FORMAT. Consequently, if the control block created last had build time errors, and a block with the same name existed in IMS.FORMAT, the block is deleted from IMS.FORMAT.

On the second pass, IMS.MFSBATCH is reprocessed together with the module table to write the new control blocks, and last occurrences of the duplicate control blocks, into IMS.FORMAT and STOW them for a directory update.

At the end of this step, the SEQBLKS data set is emptied for subsequent use by the MFSBTCH1 procedure.

JCL Requirements

The JCL for the MFSBTCH2 procedure is shown in Figure 5 on page 12. Refer to “JCL Parameter Descriptions” on page 19 for details on the EXEC parameters and DD statements.

MFS Language Utility

```
//          PROC RGN=360K,SOUT=A,COMPR2=COMPRESS,
//          COMPR3=NOCOMPRESS,DIRUPDT=UPDATE,SYS2=
//S2        EXEC PGM=DFSUNUB0,REGION=&RGN,
//          PARM='&COMPR2,&COMPR3,&DIRUPDT'
//STEPLIB DD DSN=IMS.&SYS2.RESLIB,DISP=SHR
//SEQBLKS DD DSN=IMS.&SYS2.MFSBATCH,DISP=(OLD,KEEP)
//UTPRINT DD SYSOUT=&SOUT,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYSUDUMP DD SYSOUT=&SOUT
//FORMAT DD DSN=IMS.&SYS2.FORMAT,DISP=SHR
//DUMMY DD DSN=IMS.&SYS2.PROCLIB(FMTCPY),DISP=SHR
//SYSPRINT DD SYSOUT=&SOUT
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
```

Figure 5. MFSBTCH2 Procedure

FMTCPY Control Statement: The MFSBTCH2 procedure uses this control statement to compress FORMAT.

```
COPY INDD=FORMAT,OUTDD=FORMAT
```

Test Mode

This section describes the test mode of operation using the MFSTEST procedure.

Figure 6 on page 13 shows an overview of the test mode of operation using the MFSTEST procedure. For the MFSTEST procedure JCL, see “JCL Requirements” on page 14.

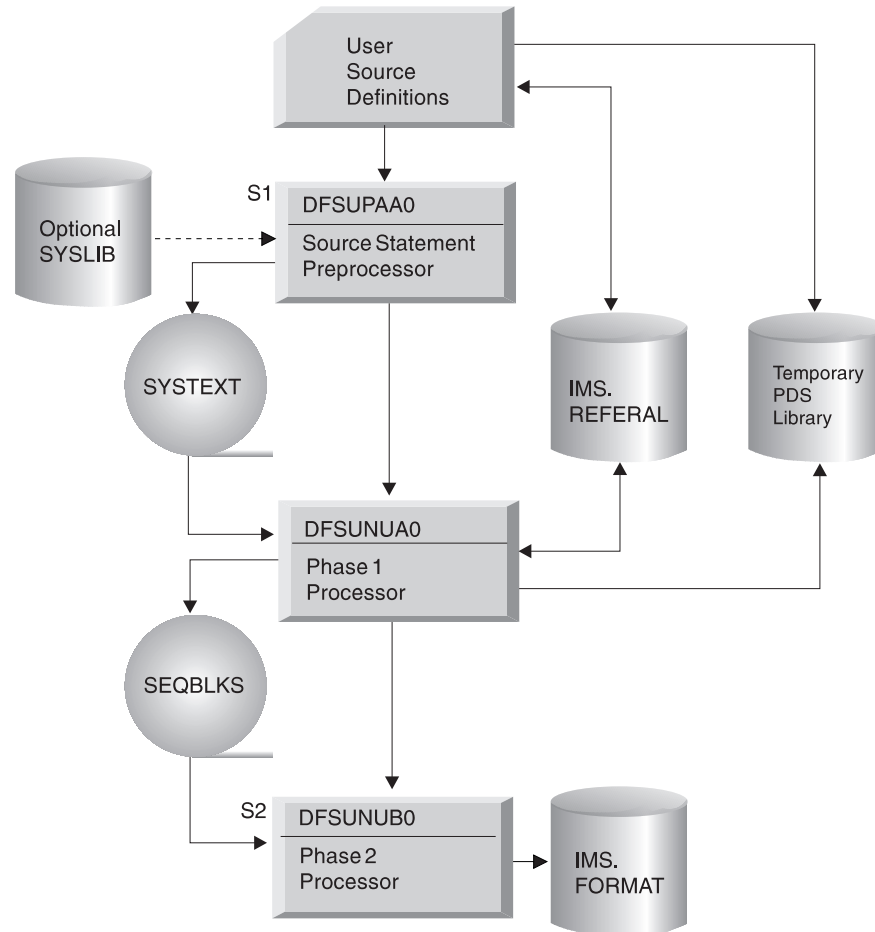


Figure 6. Overall Flow with the MFSTEST Procedure

MFSTEST Procedure Description

The MFSTEST procedure can be used to create message descriptors and device formats while an IMS control region is active. You can check the control blocks created by MFSTEST, without disrupting online production activity, by using the /TEST MFS command. Control blocks that have been tested can be placed into the staging library using the MFSUTL procedure. The MFSTEST procedure does not alter the staging library, and all of the ITBs and control blocks remain stable.

This section describes how the two-step execution of the MFS Language utility differs when the MFSTEST procedure is used.

Step 1 (S1)

Source Statement Preprocessor: The source statement preprocessor operates in the same manner as the MFSUTL procedure with the exception that the ITBs are placed into a temporary library. The contents of the historical reference library, IMS.REFERAL, are not changed to reflect new MSG, FMT, PDB, or TABLE ITBs, or new relationships that result from this test mode execution. The IMS.REFERAL library is used only in a read-only manner to perform the resolution function that ensures that all required MSG and FMT ITBs are processed.

MFS Language Utility

Phase 1 Processor: The phase 1 processor operates identically to the operation with the MFSUTL procedure, except that the ITBs for the required MSGs and FMTs are read in from the concatenated temporary library created by the preprocessor and from the IMS.REFERAL library.

The phase 1 processor obtains all MSGs, FMTs, PDBs, and TABLEs defined by this execution from the temporary library created by the preprocessor. Additional blocks, if related ITBs are present, are obtained from IMS.REFERAL.

Step 2 (S2)

Phase 2 Processor: The phase 2 processor operates against a special format library, IMS.TFORMAT, which is used by the IMS control region to access MFS control blocks when terminals are in MFSTEST mode. The phase 2 processor deletes control blocks from this library if new versions are created during this execution or if errors are detected during this execution. The phase 2 processor then inserts the new control blocks created during this execution into the library which will be available for online testing.

IMS.TFORMAT is not compressed, since the IMS control region might be concurrently reading from it.

Recommendation: It is recommended that you periodically compress this data set when the IMS control region is **not** executing (use DISP=OLD for IMS.TFORMAT).

The test procedure deletes \$\$IMSDIR, if one exists on the test format data set.

JCL Requirements

The JCL for the MFSTEST procedure is shown in Figure 7. Refer to “JCL Parameter Descriptions” on page 19 for details on the EXEC parameters and DD statements.

```
//          PROC RGN=360K,SOUT=A,SNODE=IMS,SYS2=,
//          SOR=NOLIB,MBR=NOMBR,PXREF=NOXREF,
//          PCOMP=NOCOMP,PSUBS=NOSUBS,PDIAG=NODIAG,
//          COMPR=NOCOMPRESS,LN=55,SN=8,DEVCHAR=0
//S1        EXEC PGM=DFSUPAA0,REGION=&RGN,
//          PARM=(&PXREF,&PCOMP,&PSUBS,&PDIAG,&COMPR,
//          'LINECNT=&LN,STOPRC=&SN,DEVCHAR=&DEVCHAR')
//STEPLIB DD DSN=IMS.&SYS2.RESLIB,DISP=SHR
//*SYSLIB - USER OPTION
//SYSIN DD DSN=&SNODE..&SOR.(&MBR),DISP=SHR
//REFIN DD DSN=IMS.&SYS2.REFERAL,DISP=OLD1
//REFOUT DD DSN=&&TEMPDDS,
//          DCB=IMS.&SYS2.REFERAL,
//          UNIT=SYSDA,SPACE=(CYL,(5,1,10))
//REFRD DD DSN=*.REFOUT,VOL=REF=*.REFOUT,DISP=(OLD,DELETE)
//          DD DSN=IMS.&SYS2.REFERAL,DISP=OLD1
//SYSTEXT DD DSN=&&TXTPASS,UNIT=SYSDA,;
//          SPACE=(CYL,(1,1)),DCB=BLKSIZE=800
//SYSPRINT DD SYSOUT=&SOUT,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYSUDUMP DD SYSOUT=&SOUT
//SEQBLKS DD DSN=&&BLKS,DISP=(NEW,PASS),
```

Figure 7. MFSTEST Procedure (Part 1 of 2)

```
//          UNIT=SYSDA,SPACE=(CYL,(1,1))
//S2      EXEC PGM=DFSUNUB0,REGION=&RGN,
//          PARM='TEST,DEVCHAR=&DEVCHAR',
//          COND=(8,LT,S1)
//STEPLIB DD   DSN=IMS.&SYS2.RESLIB,DISP=SHR
//SEQBLKS DD   DSN=&&BLKS,DISP=(OLD,DELETE)
//UTPRINT DD   SYSOUT=&SOUT,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYSUDUMP DD   SYSOUT=&SOUT
//FORMAT  DD   DSN=IMS.&SYS2.TFORMAT,DISP=SHR
```

Figure 7. MFSTEST Procedure (Part 2 of 2)

The DISP=OLD specifications are required.

Restriction: A DD DUMMY specification is not supported.

MFS Library Backup and Restore Operations

This section describes the two procedures, MFSBACK and MFSREST, which are provided to perform utility library backup and restore operations. The JCL required for the MFSBACK and MFSREST procedures are shown below.

Attention: When you use these procedures, make sure that the IMS.REFERAL and IMS.FORMAT libraries are dumped and restored at the same level, that is, at the same time. It is important to do this because of the relational information in the IMS.REFERAL PDS directory which describes the contents of the libraries. To ensure that all libraries are restored to the same level, scratch and reallocate all MFS data sets prior to performing the restore operation. If the libraries are not restored to the same level, unpredictable operation can occur.

MFSBACK Procedure

Figure 8 on page 16 shows the JCL for the MFSBACK procedure and includes the optional MFSTEST facility. All DISP=OLD specifications are required.

Restriction: A DD DUMMY specification is not supported in the statements that require DISP=OLD.

The block size for the IMS.REFERAL library, if specified, must be 800.

Refer to “JCL Parameter Descriptions” on page 19 for details on the EXEC parameters and DD statements.

MFS Language Utility

```
//          PROC  NODE=IMS,TAPE=MFSDBS,SOUT=A,
//          DSN=FORMAT,SYS2=
//*
//*****
//*
//*  PROCEDURE KEYWORDS FOR // EXEC STATEMENT:
//*
//*  NODE=  PREFIX LEVEL TO BE USED FOR
//*         ACCESS TO IMS MFS LIBRARIES.
//*
//*  SYS2=  SECOND PREFIX LEVEL TO BE USER FOR
//*         ACCESS TO IMS MFS LIBRARIES.
//*
//*  TAPE=  BACKUP TAPE SERIAL NUMBER.
//*
//*  SOUT=  SPECIFIES THE PRINT OUTPUT CLASS
//*         TO BE USED FOR PRINTED OUTPUT
//*         DURING THE BACKUP OPERATION.
//*
//*****
//*
//MOVE1 EXEC  PGM=IEBCOPY,PARM='SIZE=100K'
//SYSPRINT DD  SYSOUT=&SOUT
//SYSUT3 DD   SPACE=(CYL,(1,1)),UNIT=SYSDA
//REFERAL DD  DSN=&NODE..&SYS2.REFERAL,DISP=OLD
//TAPEOUT DD  UNIT=2400,LABEL=(1,SL),DISP=(NEW,PASS),
//            VOL=(,RETAIN,SER=&TAPE),
//            DSN=&NODE..&SYS2.REFERAL,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//*
//*****
//*
//*  //MOVE1.SYSIN DD * MUST BE SUPPLIED BY THE
//*  USER WITH THE APPROPRIATE COPY CONTROL
//*  STATEMENT AS SHOWN BELOW:
//*
//*  COPY OUTDD=TAPEOUT,INDD=REFERAL
//*
//*****
//*
//MOVE2 EXEC  PGM=IEBCOPY,PARM='SIZE=100K'
//SYSPRINT DD  SYSOUT=&SOUT
//SYSUT3 DD   UNIT=SYSDA,SPACE=(CYL,(1,1))
//FORMAT DD   DSN=&NODE..&SYS2.&DSN,DISP=OLD;
//TAPEOUT DD  UNIT=2400,LABEL=(2,SL),
//            VOL=(,RETAIN,REF=*.MOVE1.TAPEOUT),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
//            DSN=&NODE..&SYS2.&DSN,
//            DISP=(OLD,KEEP)
//*
//*****
//*  //MOVE2.SYSIN DD * MUST BE SUPPLIED WITH
//*  APPROPRIATE COPY CONTROL STATEMENT
//*  AS SHOWN BELOW:
//*
//*  COPY OUTDD=TAPEOUT,INDD=FORMAT
//*
//*****
//*
```

Figure 8. MFSBACK Procedure (Part 1 of 2)

```

//MOVE3 EXEC PGM=IEBCOPY,PARM='SIZE=100K'
//SYSPRINT DD SYSOUT=&SOUT
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//FORMAT DD DSN=&NODE..&SYS2.TFORMAT,DISP=OLD
//TAPEOUT DD UNIT=2400,LABEL=(3,SL),
//          VOL=REF=*.MOVE1.TAPEOUT,
//          DSN=&NODE;.&SYS2.TFORMAT,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
//          DISP=(OLD,KEEP)
//*
//*****
//* //MOVE3.SYSIN DD * MUST BE SUPPLIED WITH *
//* THE APPROPRIATE COPY CONTROL STATEMENT *
//* AS SHOWN BELOW: *
//* *
//* COPY OUTDD=TAPEOUT,INDD=FORMAT *
//* *
//*****
//*

```

Figure 8. MFSBACK Procedure (Part 2 of 2)

MFSREST Procedure

Figure 9 on page 18 shows the JCL for the MFSREST procedure and includes the optional MFSTEST facility. All DISP=OLD specifications are required.

Restriction: A DD DUMMY specification is not supported in the statements that require DISP=OLD.

Refer to “JCL Parameter Descriptions” on page 19 for details on the EXEC parameters and DD statements.

MFS Language Utility

```
//          PROC  NODE=IMS,SYS2=,TAPE=MFSDBS,SOUT=A,
//          DSN=FORMAT
//*
//*****
//*
//*  PROCEDURE KEYWORDS FOR // EXEC STATEMENT:
//*
//*  NODE=  PREFIX LEVEL TO BE USED FOR
//*         ACCESS TO IMS MFS LIBRARIES.
//*
//*  SYS2=  SECOND PREFIX LEVEL TO BE USED FOR
//*         ACCESS TO IMS MFS LIBRARIES.
//*
//*  TAPE=  RESTORE TAPE SERIAL NUMBER.
//*
//*  SOUT=  SPECIFIES THE PRINT OUTPUT CLASS
//*         TO BE USED FOR PRINTED OUTPUT
//*         DURING THE RESTORE OPERATION.
//*
//*****
//*
//MOVE1 EXEC  PGM=IEBCOPY,PARM='SIZE=100K'
//SYSPRINT DD  SYSOUT=&SOUT
//SYSUT3 DD  SPACE=(CYL,(1,1)),UNIT=SYSDA
//REFERAL DD  DSN=&NODE..&SYS2.REFERAL,DISP=OLD
//TAPEIN DD  UNIT=2400,LABEL=(1,SL),DISP=(OLD,KEEP),
//           VOL=(,RETAIN,SER=&TAPE),
//           DSN=&NODE..&SYS2.REFERAL,
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//*
//*****
//*
//*  //MOVE1.SYSIN DD * MUST BE SUPPLIED BY THE
//*  USER WITH THE APPROPRIATE COPY CONTROL
//*  STATEMENT AS SHOWN BELOW:
//*
//*  COPY OUTDD=REFERAL,INDD=TAPEIN
//*
//*****
//*//MOVE2 EXEC  PGM=IEBCOPY,PARM='SIZE=100K'
//SYSPRINT DD  SYSOUT=&SOUT
//SYSUT3 DD  UNIT=SYSDA,SPACE=(CYL,(1,1))
//FORMAT DD  DSN=&NODE..&SYS2.&DSN,DISP=OLD;
//TAPEIN DD  UNIT=2400,LABEL=(2,SL),
//           VOL=(,RETAIN,REF=*.MOVE1.TAPEIN),
//           DSN=&NODE..&SYS2.&DSN,
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
//           DISP=(OLD,KEEP)
//*
//*****
//*  //MOVE2.SYSIN DD * MUST BE SUPPLIED WITH
//*  THE APPROPRIATE COPY CONTROL STATEMENT
//*  AS SHOWN BELOW:
//*
//*  COPY OUTDD=FORMAT,INDD=TAPEIN
//*
//*****
//*
```

Figure 9. MFSREST Procedure (Part 1 of 2)

```

//MOVE3 EXEC PGM=IEBCOPY,PARM='SIZE=100K'
//SYSPRINT DD SYSOUT=&SOUT
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//FORMAT DD DSN=&NODE..&SYS2.TFORMAT,DISP=OLD
//TAPEIN DD UNIT=2400,LABEL=(3,SL),
//          VOL=REF=*.MOVE1.TAPEIN,
//          DSN=&NODE..&SYS2.TFORMAT,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
//          DISP=(OLD,KEEP)
//
//*****
//* //MOVE3.SYSIN DD * MUST BE SUPPLIED WITH *
//* THE APPROPRIATE COPY CONTROL STATEMENT *
//* AS SHOWN BELOW: *
//* *
//* COPY OUTDD=FORMAT,INDD=TAPEIN *
//* *
//*****
//*

```

Figure 9. MFSREST Procedure (Part 2 of 2)

JCL Parameter Descriptions

This section describes the JCL parameters for MFSUTL, MFSBTCH1, and MFSTEST.

When Step 1 (S1) executes (in the MFSUTL, MFSBTCH1, and MFSTEST procedures), the following parameters can be specified in the PARM keyword of the EXEC statement.

PXREF= NOXREF | XREF

Specifies whether (XREF) or not (NOXREF) a sorted cross reference listing should be provided. The default value is NOXREF. A sorted cross reference listing includes a list of all labels and related references.

PCOMP= NOCOMP | COMP

Specifies whether (COMP or COMPOSITE) or not (NOCOMP) the composite or final version of the statement, after error recovery or substitution has modified it, is printed. The default value is NOCOMP. The composite statement reflects syntactic assumptions made during error recovery. Semantic assumptions do not appear in the composite statement but are reflected in the intermediate text blocks. If the repetitive generation function for MFLD/DFLD statements is used, COMP also causes the generated statements to be printed; NOCOMP suppresses this printing.

PSUBS= NOSUBS | SUBS

Specifies whether (SUBS or SUBSTITUTE) or not (NOSUBS) the substitution variable and its equated value are printed when the substitution variable is encountered in the operand field of a statement. The default value is NOSUBS.

PDIAG= NODIAG | DIAG

Specifies whether (DIAG or DIAGNOSTIC) or not (NODIAG) the XREF, COMP, and SUBS options should all be set on. In addition, diagnostic information is printed. The default value is NODIAG, which has no effect on the XREF, COMP, and SUBS options but suppresses printing of the diagnostic information.

MFS Language Utility

COMPR= NOCOMPRESS | COMPRESS

Specifies whether (COMPRESS) or not (NOCOMPRESS) the IMS.REFERAL library is to be compressed before new ITBs are added. The default value is NOCOMPRESS.

LN= 55 | nn

Specifies how many lines per page should be printed. The default value is 55.

SN= 08 | nn

Specifies the severity code compare value. MSG, FMT, and TABLE blocks whose error severity equals or exceeds this value are not written to the IMS.REFERAL library. The default value is 08.

DEVCHAR= 0 | x

Specifies the alphanumeric suffix character (x) to be appended to DFSUDT0. The name DFSUDT0 identifies the desired device characteristics table. This suffix character (x) corresponds to the value specified in the SUFFIX= keyword of the MSGEN macro. The default is zero (0).

In the execution of the MFSRVC procedure, one parameter can be specified. The DEVCHAR=0 or x parameter specifies the alphanumeric suffix character (x) to be used for the device characteristics table, when no suffix is specified in the LIST control statement parameter DEVCHAR. The default is zero.

In the execution of Step 2 (S2) in the MFSUTL and MFSBTCH2 procedures, three parameters can be specified in the EXEC statement's PARM keyword:

COMPR2= COMPRESS | NOCOMPRESS

Specifies whether (COMPRESS) or not (NOCOMPRESS) the IMS.FORMAT library is to be compressed before new control blocks are added. The default value is COMPRESS.

COMPR3= COMPREND | NOCOMPREND

Specifies whether (COMPREND) or not (NOCOMPREND) the data set with the ddname of FORMAT is compressed after all format blocks have been added/replaced and the index directory (\$\$IMSDIR) has been updated.

DIRUPDT= UPDATE | NOUPDATE

Specifies whether (UPDATE) or not (NOUPDATE) the special index directory (\$\$IMSDIR) is automatically updated after a block has been deleted from a format library. You can bypass the \$\$IMSDIR update by specifying NOUPDATE. The default is UPDATE.

In the execution of Step 2 (S2) in the MFSTEST procedure, the PARM='TEST' parameter must be specified.

Other EXEC statement parameters that can be specified are:

RGN=

Specifies the region size for this execution. The default is 360K.

SOUT=

Specifies the SYSOUT class. The default is A.

SNODE=

Specifies the node that can be assigned to the MFS utility data set name. The default value is IMS.

SOR=

Specifies the library name that can be assigned to the MFS utility library for SYSIN or SYSLIB. The default value is NOLIB.

MBR=

Specifies the member name that can be assigned to the MFS utility member for SYSIN. The default is NOMBR.

DDNAMES Used in MFS Procedures

The data set names used in the MFSUTL, MFSBTCH1, MFSBTCH2, and MFSTEST procedures fit installation needs. The ddnames used and the data sets they refer to are:

REFIN**REFOUT****REFRD**

Refers to the MFS reference library, except when used in the MFSTEST procedure. In MFSTEST, REFIN and REFRD refer to the MFS reference library; REFOUT is a temporary data set.

FORMAT

Refers to the MFS control block library. In MFSTEST, this ddname refers to the MFS test control block library.

SYSLIB

Refers to an optional user library from which input can be copied.

SYSIN

Refers to the input data set, which can be a sequential data set or a member of a partitioned data set.

DUMMY

Refers to the IMS procedure library, which contains control statements used to compress the MFS reference and control block libraries.

SYSUT3**SYSUT4**

Are ddnames for data sets used during the data set compression as work data sets.

DUMMY, SYSUT3, and SYSUT4 can all be omitted if neither the MFS reference library nor the MFS control block library is to be compressed.

UTPRINT

Is used for messages during the compression of the MFR reference library, and is used for MFS error and status messages during MFS Language utility Phase 2 processing.

The following ddnames refer to data sets used in the MFSRVC procedure. The data set names can be altered to fit installation needs.

REFIN

Refers to the MFS reference library.

FORMAT

Refers to the MFS control block library.

SYSIN

Refers to the input data set, which can be a sequential data set or a member of a partitioned data set.

MFS Language Utility

SYSSNAP

Refers to a data set that is used to receive the output from a SNAP macro if certain severe errors are detected.

SYSPRINT

Refers to the destination of the output. If output is to be sent to a data set (instead of SYSOUT=), use DISP=MOD for the data set.

MFSUTL and MFSTEST Region Parameter Estimate

The following steps help you estimate the main storage requirements that you should specify in the RGN= parameter of the EXEC statement invoking the MFSUTL and MFSTEST procedures.

1. Calculate statement base count. For the input to the MFS Language utility, determine the largest (number of statements) device format to be processed and the largest message descriptor related to the format. Add the total number of statements contained in these two control blocks to obtain the statement base count.

For the processing of a specific user-supplied MSG or FMT ITB, the utility reprocesses all related MSG or FMT ITBs saved from the IMS.REFERAL data set to ensure compatible linkage between all related online blocks. These reprocessed ITBs must be analyzed as well for the above process of obtaining the statement base count.

2. Estimate Region Requirements. Multiply the statement base count by 214 and add 300000 to the result. Round the resulting value to the next highest multiple of 2048. The result is an estimate of the main storage requirements which should be specified in the RGN= parameter of the EXEC statement invoking the MFSUTL and MFSTEST procedures.

Complex formats with a large number of literal DFLD statements in relation to the statement base count can exceed the estimate.

Chapter 2. MFS Device Characteristics Table Utility (DFSUTB00)

The Message Format Service Device Characteristics Table (MFSDCT) utility (DFSUTB00) defines new screen sizes in a descriptor member of the IMS.PROCLIB library without performing an IMS system generation. These new screen size definitions are added to the screen sizes that were previously defined.

The MFSDCT (DFSUTB00) utility procedure consists of the following steps:

1. The DFSUTB00 program is executed to initiate several functions.

The DFSUTB00 program:

- Reads one or two descriptor members from PROCLIB and uses only the new device descriptors as input.
 - Builds DCTENTRY statements for each device descriptor.
 - Optionally loads an existing device characteristics table from JOBLIB/STEPLIB data sets (usually from the IMS.RESLIB library) and then builds DCENTRY statements for each DCT entry.
 - Invokes the assembler, passing the DCTENTRY statements and the DCTBLD and MFSINIT macros as input.
 - Reads the output from the assembler as an updated or new device characteristics table and as a new set of default MFS format definitions. (This output is split into separate files for later processing.)
2. The assembler is invoked to assemble the new device characteristics table.
 3. The linkage editor is invoked to link-edit the new device characteristics table into the IMS.RESLIB.
 4. Phase 1 of the MFS Language utility generates new default MFS format control blocks.
 5. Phase 2 of the MFS Language utility puts the new default MFS format control blocks into the IMS.FORMAT library.

In this Chapter:

- “Restrictions”
- “MFSDCT Procedure”

Restrictions

The following restrictions apply to this utility:

- The utility ignores all other descriptors while reading the one or two descriptor members from PROCLIB.
- At least one device descriptor must be specified or the utility terminates.

MFSDCT Procedure

This section explains the JCL for the MFSDCT procedure, the PROC statement, the EXEC statement, the DD statements, the MFS device descriptions, and error processing.

Figure 10 on page 24 shows the JCL for the procedure used to invoke the MFS Device Characteristics Table utility.

MFSDCT Utility

```
//          PROC RGN=4M,SOUT=A,SYS2=,PXREF=NOXREF,
//          PCOMP=NOCOMP,PSUBS=NOSUBS,PDIAG=NODIAG,
//          COMPR=NOCOMPRESS,COMPR2=COMPRESS,
//          LN=55,SN=8,DEVCHAR=0,COMPR3=NOCOMPRESS,
//          DIRUPDT=UPDATE,DCTSUF=,
//          DSCTSUF=,DSCMSUF=,FMTCMAST=N
//S1        EXEC PGM=DFSUTB00,REGION=&RGN,
//          PARM=('DCTSUF=&DCTSUF,DSCTSUF=&DSCTSUF',
//          'DSCMSUF=&DSCMSUF,DEVCHAR=&DEVCHAR,FMTCMAST=&FMTCMAST')
//STEPLIB  DD DSN=IMS.&SYS2.RESLIB,DISP=SHR
//SYSLIB   DD DSN=IMS.OPTIONS,DISP=SHR
//          DD DSN=IMS.GENLIB,DISP=SHR
//          DD DSN=IMS.GENLIBA,DISP=SHR
//          DD DSN=IMS.GENLIBB,DISP=SHR
//          DD DSN=SYS1.MACLIB,DISP=SHR
//          DD DSN=SYS1.MODGEN,DISP=SHR
//PROCLIB  DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
//SYSIN    DD DSN=&&SYSIN,UNIT=SYSDA,;
//          SPACE=(CYL,(1,1)),DCB=BLKSIZE=800
//SYSPUNCH DD DSN=&&SYSPUNCH,UNIT=SYSDA,;
//          SPACE=(CYL,(1,1)),DCB=BLKSIZE=800
//SYSUT1   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSPRINT DD SYSOUT=&SOUT,
//          DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYSUDUMP DD SYSOUT=&SOUT
//DCTIN    DD DSN=&&DCTIN,DISP=(NEW,PASS),
//          UNIT=SYSDA,SPACE=(CYL,(1,1))
//DEFLT    DD DSN=&&DEFLT,DISP=(NEW,PASS),
//          UNIT=SYSDA,SPACE=(CYL,(1,1))
//DCTLNK   DD DSN=&&DCTLNK,DISP=(NEW,PASS),
//          UNIT=SYSDA,SPACE=(TRK,(1,1)),
//          DCB=BLKSIZE=800
//S2        EXEC PGM=ASMA90,REGION=& RGN,
//          PARM='OBJECT,NODECK,NOLIST',
//          COND=(0,LT)
//SYSLIB   DD DSN=IMS.OPTIONS,DISP=SHR
//          DD DSN=IMS.GENLIB,DISP=SHR
//          DD DSN=IMS.GENLIBA,DISP=SHR
//          DD DSN=IMS.GENLIBB,DISP=SHR
//          DD DSN=SYS1.MACLIB,DISP=SHR
//          DD DSN=SYS1.MODGEN,DISP=SHR
//SYSLIN   DD DSN=&&DCT,DISP=(NEW,PASS)
//          UNIT=SYSDA,SPACE=(CYL,(1,1)),
//          DCB=BLKSIZE=800<
//SYSPRINT DD SYSOUT=&SOUT,
//          DCB=(BLKSIZE=605),
//          SPACE=(605,(100,50),RLSE,,ROUND)
//SYSUT1   DD UNIT=SYSDA,DISP=(,DELETE),
//          SPACE=(CYL,(15,15))
//SYSIN    DD DSN=&&DCTIN,DISP=(OLD,DELETE)
//S3        EXEC PGM=IEWL,
//          PARM=('SIZE=(880K,64K)',NCAL,LET,REUS,
//          XREF,LIST),
//          REGION=&RGN,
//          COND=(0,LT)
//SYSPRINT DD SYSOUT=&SOUT,
//          DCB=(RECFM=FBA,LRECL=121,BLKSIZE=605),
//          SPACE=(605,(10,10),RLSE,,ROUND)
```

Figure 10. MFSDCT Procedure (Part 1 of 2)

```

//SYSLMOD DD DSN=IMS.&SYS2.RESLIB,DISP=SHR
//SYSUT1 DD UNIT=(SYSDA,SEP=(SYSLMOD,SYSPUNCH)),
//        SPACE=(CYL,(10,1))
//SYSLIN DD DSN=&&DCTLNK,DISP=(SHR,DELETE)
//DCT DD DSN=&&DCT,DISP=(SHR,DELETE)
//S4 EXEC PGM=DFSUPAA0,REGION=&RGN,
//      PARM=(&PXREF,&PCOMP,&PSUBS,&PDIAG,;
//      &COMPR,'LINECNT=&LN,STOPRC=&SN',
//      'DEVCHAR=&DEVCHAR'),COND=(0,LT)
//STEPLIB DD DSN=IMS.&SYS2.RESLIB,DISP=SHR
//*SYSLIB - USER OPTION
//SYSIN DD DSN=&&DEFLT,DISP=(OLD,DELETE)
//REFIN DD DSN=IMS.&SYS2.REFERAL,DISP=OLD
//REFOUT DD DSN=IMS.&SYS2.REFERAL,DISP=OLD
//REFRD DD DSN=IMS.&SYS2.REFERAL,DISP=OLD
//SYSTEXT DD DSN=&&TXTPASS,UNIT=SYSDA,;
//        SPACE=(CYL,(1,1)),DCB=BLKSIZE=800
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//DUMMY DD DISP=SHR,
//        DSN=IMS.&SYS2.PROCLIB(REFCPY)
//UTPRINT DD SYSOUT=&SOUT
//SYSPRINT DD SYSOUT=&SOUT,
//        DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYSUDUMP DD SYSOUT=&SOUT
//SEQBLKS DD DSN=&&BLKS,DISP=(NEW,PASS),
//        UNIT=SYSDA,SPACE=(CYL,(1,1))
//S5 EXEC PGM=DFSUNUB0,REGION=&RGN,
//      PARM=(&COMPR2,&COMPR3,&DIRUPDT,;
//      'DEVCHAR=&DEVCHAR'),COND=((0,LT,S1),
//      (0,LT,S2),(0,LT,S3),(8,LT,S4))
//STEPLIB DD DSN=IMS.&SYS2.RESLIB,DISP=SHR
//SEQBLKS DD DSN=&&BLKS,DISP=(OLD,DELETE)
//UTPRINT DD SYSOUT=&SOUT,
//        DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYSUDUMP DD SYSOUT=&SOUT
//FORMAT DD DSN=IMS.&SYS2.FORMAT,DISP=SHR
//DUMMY DD DISP=SHR,
//        DSN=IMS.&SYS2.PROCLIB(FMTCPY)
//SYSPRINT DD SYSOUT=&SOUT
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,(1,1))

```

Figure 10. MFSDCT Procedure (Part 2 of 2)

PROC Statement

The procedure statement must be in the following form:

```

PROC RGN=4M,SOUT=A,SYS2=,PXREF=NOXREF,
PCOMP=NOCOMP,PSUBS=NOSUBS,PDIAG=NODIAG,
COMPR=NOCOMPRESS,COMPR2=COMPRESS,
LN=55,SN=8,DEVCHAR=0,COMPR3=NOCOMPRESS,
DIRUPDT=UPDATE,DCTSUF=,
DSCTSUF=,DSCMSUF=

```

In addition to the optional keyword parameters described in “JCL Parameter Descriptions” on page 19, the following parameters can be specified. (x is the alphanumeric suffix character that you are appending to the member name.)

DCTSUF=x

Specifies the suffix character to be appended to DFSUDT0. The name DFSUDT0x identifies the device characteristics table to which new definitions are added. This suffix character corresponds to the value specified in the

MFSDCT Utility

SUFFIX= keyword of the IMSGEN macro. If a suffix character is not specified, a completely new device characteristics table is built from just the device descriptors.

DSCTSUF=x

Specifies the suffix character to be appended to DFSDSCT. The name DFSDSCTx identifies a descriptor member. This suffix character corresponds to the value specified in the IMS procedure DSCT= keyword. This parameter is required if DSCMSUF= is not specified.

DSCMSUF=x

Specifies the suffix character to be appended to DFSDSCM. The name DFSDSCMx identifies a descriptor member. This suffix character corresponds to the value specified in the SUFFIX= keyword of the IMSGEN macro. This parameter is required if DSCTSUF= is not specified.

DEVCHAR=0 | x

Specifies the suffix character to be appended to DFSUDT0. The name DFSUDT0x identifies the updated or new device characteristics table. Any existing device characteristics table with the same name is replaced in IMS.RESLIB by step 3 of this utility. The default is 0.

FMTMAST=Y/N

Specifies whether (Y) or not (N) the IMS-provided support for MFS on the master terminal is to be used.

EXEC Statement

The EXEC statement determines that a device characteristics table is created. It also specifies the name for the desired descriptor member and the name of the updated or new device characteristics table. Each of the five steps in this procedure names a different program for execution.

The format for each step is shown in Figure 11.

```
S1      EXEC PGM=DFSUTB00,REGION=&&RGN,
        PARM=('DCTSUF=&&DCTSUF,DSCTSUF=&&DSCTSUF'
        'DSCMSUF=&&DSCMSUF,DEVCHAR=&&DEVCHAR')
S2      EXEC PGM=ASMA90,REGION=&&RGN,
        PARM=('OBJECT,NODECK,NOLIST',
        COND=(0,LT)')
S3      EXEC PGM=IEWL,
        PARM=('SIZE=880K,64K),NCAL,LET,REUS,XREF,LIST',
        REGION=&&RGN,
        COND=(0,LT)
S4      EXEC PGM=DFSUPAA0,REGION=&&RGN,
        PARM=(&&PXREF,&&PCOMP,&&PSUBS,&&PDIAG,&&COMPR,;
        'LINECNT=&&LN,STOPRC=&&SN,DEVCHAR=&&DEVCHAR'),
        COND=(0,LT)
S5      EXEC PGM=DFSUNUB0,REGION=&&RGN,
        PARM=(&&COMPR2,&&COMPR3,&&DIRUPDT,;
        'DEVCHAR=&&DEVCHAR'),COND=((0,LT,S1),
        (0,LT,S2),(0,LT,S3),(8,LT,S4))
```

Figure 11. Five Steps of the MFSDCT (DFSUTB00) Utility

In addition to the optional keyword parameters described in “JCL Parameter Descriptions” on page 19, you can also specify the parameters described in “PROC Statement” on page 25.

DD Statements

The following ddnames are used in step 1 of the MFSDCT procedure.

DCT

Defines the temporary data set for the updated or new device characteristics table as output from the assembler with the ddname SYSLIN (step 2) and as input to the Linkage Editor with ddname DCT (step 3).

DCTIN

Defines a temporary data set for the device characteristics table as input to the assembler (step 2).

DCTLNK

Defines the temporary data set for the link-edit control statements for step 3.

DEFLTS

Defines the temporary data set for the default MFS format definitions for MFS Language utility input (step 4).

PROCLIB

Defines the libraries containing the descriptor members DFSDSCMx and DFSDSCTx.

STEPLIB

Defines the libraries containing the program DFSUTB00 and the device characteristics table specified in the DCTSUF= parameter.

SYSIN

Defines the temporary file containing the generated DCENTRY statements.

SYSLIN

Defines the temporary data set for the updated or new device characteristics table as output from the assembler (step 2) and as input to the Linkage Editor with ddname DCT (step 3).

SYSLIB

Defines the libraries containing IMS and MVS/ESA macros.

SYSPRINT

Defines the data set for all of the printed output from step 1, including error messages and output from steps 2 and 3.

SYSPUNCH

Defines the temporary file containing the object module output from the assembler. The output is the device characteristics table, followed immediately by the default MFS format definitions.

SYSUT1

Defines an assembler and linkage editor work data set.

SYSMOD

Defines the IMS.RESLIB data set to contain the new or modified device characteristics table.

MFS Device Descriptors

MFS device descriptors are used by the MFS Device Characteristics Table utility to update screen size in the DCT and generate new MFS default formats without system generation.

Related Reading: See *IMS/ESA Administration Guide: Transaction Manager* for more information about how to use ETO descriptors.

MFSDCT Utility

MFS Device Descriptor Format

The format for an MFS device descriptor is:

D descriptor name parm1 parm2 parm3

D Is the descriptor type.

descriptor name

Is ignored.

parm (1...3)

Are the keywords TYPE=, SIZE=, and FEAT=.

Related Reading: See *IMS/ESA Installation Volume 2: System Definition and Tailoring* for information about the values for SIZE, TYPE, and FEAT on the TERMINAL macro.

Error Processing

Return codes are based on the error message.

Related Reading: See *IMS/ESA Messages and Codes* for explanations of the return code and error message.

Part 2. Service Utilities

Chapter 3. MFS Service Utility (DFSUTSA0)	31
Restrictions	32
MFSRVC Procedure	32
PROC Statement	32
EXEC Statement	33
DD Statements	33
Invoking the Procedure	33
Function Descriptions	33
INDEX	34
DELETE	34
SCRATCH	35
RELATE	37
Sample Output	37
LIST	37
Sample Output	39
LIST the MFS Device Characteristics Table	40
Example 1	40
Example 2	40
LIST DEVCHAR Output	41
Sample output	41
Utility Control Statements	41
Positional Parameters	41
Keywords	42
 Chapter 4. Multiple Systems Verification Utility (DFSUMSV0)	45
Restrictions	45
Prerequisites	46
IMSMSV Procedure	46
PROC Statement	47
EXEC Statement	47
Invoking the Procedure	47
Input	48
Input Validation	48
Multisystem Control Block Verification	48
Partner IDs and Assigned Physical Links	48
Remote SYSID to Local SYSID Paths	49
Remote and Local Transaction Attributes	49
Presence of Corresponding Logical Terminals	50
Output Messages and Path Map	50
Utility Control Statements	52
Error Processing	53
 Chapter 5. Spool SYSOUT Print Utility (DFSUPRT0)	55
Restrictions	55
Input and Output	55
DFSWTnnn Procedure	55
PROC Statement	56
EXEC Statement	56
DD statements	56
IMSWTnnn Job	57
Sample Output	57
Error Processing	57

Chapter 6. Time-Controlled Operations Verification Utility (DFSTVER0)	59
TCO Verification Procedure	59
EXEC Statement	60
DD Statements	61
Output	61
Error Report	61
Statistics Report	62
Timer-Elements Report	62
Message-Table Report	63
Summary Report	63
Return Codes	64

Chapter 3. MFS Service Utility (DFSUTSA0)

The MFS Service (MFSRVC) utility (DFSUTSA0) helps to control and maintain MFS intermediate text blocks and control blocks once they have been processed and stored by the MFS Language utility. An intermediate text block (ITB) is a message, format, partition set, or table source language definition stored in the IMS.REFERAL library. A control block is a message or format definition stored in the IMS.FORMAT, IMS.FORMATA, IMS.FORMATB, or IMS.TFORMAT library.

The service utility performs the following functions:

- INDEX creates a special directory for faster access to IMS.FORMAT control blocks.
- DELETE deletes specified contents of the special index directory (\$\$IMSDIR).
- SCRATCH scratches specified contents of the IMS.FORMAT and IMS.REFERAL libraries and their directories (SCRATCH also operates on IMS.TFORMAT).
- RELATE produces an interpreted listing of the contents of the IMS.REFERAL library.
- LIST produces an interpreted listing of either:
 - The contents of the IMS.FORMAT or IMS.TFORMAT library
 - The contents of the special index directory in the IMS.FORMAT library
 - The contents of the MFS device characteristics table (DFSUDT0x) in the IMS.RESLIB library.

The error message

ERR TYPE x IN REFERAL LIBRARY, FUNCT=RELATE

is issued if a problem exists in the MFS REFERAL library. The error types are:

- | | |
|----------|------------------------------|
| 1 | Directory block length error |
| 2 | User appendage length error |
| 3 | Unknown block error |
| 4 | Duplicate DUMMY FORMAT error |
| 5 | Duplicate FORMAT error |
| 6 | TABLE block error |
| 7 | Message input block error |
| 8 | REFIN OPEN error |
| 9 | REFIN OPEN SYNAD taken |
| A | PDB block error |
| B | FORMAT block error |
| C | Message output block error |

If an error is detected in the SYSIN, SYSPRINT, or REFIN <REFERAL> file, a return code is set to 4, 8 or 12 respectively.

If an error is detected on the RELATE function, the subsequent functions are ignored.

In this Chapter:

- “Restrictions”
- “MFSRVC Procedure”
- “Function Descriptions” on page 33
- “Utility Control Statements” on page 41

Restrictions

The following restrictions apply to this utility:

- Do not run the MFS Service utility concurrently with the MFS Language utility (MFSUTL procedure) if they are accessing the same data set.
- Do not run the MFS Service utility concurrently with the IMS online control region if they are both accessing the active format library.

MFSRVC Procedure

This section describes the JCL for the utility, the PROC statement, the EXEC statement, the DD statements, and invoking the procedure.

Figure 12 shows a one-step procedure for maintaining the MFS libraries. It is placed in the IMS.PROCLIB by Stage 2 of system generation.

```
//          PROC DEVCHAR=0,SYS2=,SOUT=A
//MFSRVC   EXEC PGM=DFSUTSA0,REGION=250K,PARM='DEVCHAR=&DEVCHAR'
//STEPLIB  DD   DSN=IMS.&SYS2.RESLIB,DISP=SHR
//*
//*        PRINT FILES
//*
//SYSPRINT DD   SYSOUT=&SOUT
//*        DCB=(RECFM=VBA,LRECL=137)
//SYSSNAP  DD   SYSOUT=&SOUT
//*        DCB=(RECFM=VBA,LRECL=125,BLKSIZE=1632)
//SYSUDUMP DD   SYSOUT=&SOUT
//*
//*        REFERRAL LIBRARY
//*
//REFIN    DD   DSN=IMS.&SYS2.REFERAL,DISP=OLD
//*
//*        ON-LINE FORMAT LIBRARY
//*
//FORMAT   DD   DSN=IMS.&SYS2.FORMAT,DISP=SHR
//*
//*
//*        //SYSIN DD * MUST BE SUPPLIED BY
//*        USER WITH INPUT CONTROL CARD STREAM
//*
//*        ALL DISP=OLD SPECIFICATIONS OF THIS
//*        PROCEDURE ARE REQUIRED .....
//*
```

Figure 12. MFSRVC Procedure

PROC Statement

The PROC statement must be in the form:

DEVCHAR=0,SYS2=,SOUT=A

DEVCHAR=

Specifies the device characteristics table. The default is 0.

SOUT=

Specifies the SYSOUT class. The default is A.

SYS2=

Specifies an optional second level dsname qualifier for those data sets which are designated as "Optional Replicate" in an XRF complex. When specified, the operand must be enclosed in quotes and must include a trailing period; for example, SYS2='IMSA.'.

EXEC Statement

The EXEC statement must be in the form:

```
PGM=DFSUTSA0,REGION=250K,PARM='DEVCHAR=&DEVCHAR'
```

REGION=

Specifies the region size for the execution of the MFS Service utility. The default is 250K.

PARM=

The PARM= field must be in the form:

```
PARM='DEVCHAR=&DEVCHAR'
```

where &DEVCHAR is the device characteristics table to be listed.

DD Statements

STEPLIB DD

Points to IMS.RESLIB, which contains the IMS nucleus and required action modules.

SYSPRINT DD

Defines the output message data set. It can be a printer or it can be routed through the output stream. If DISP=(MOD,...) is specified, it can be a tape volume or a direct-access device. The output can be blocked as a multiple of 121.

SYSSNAP

Refers to a data set that is used to receive the output from a SNAP macro if certain severe errors are detected.

Invoking the Procedure

The following JCL statements are used to invoke the MFSRVC procedure.

```
//MFSRVC JOB MSGLEVEL=1
//      EXEC MFSRVC
//SYSIN DD *
      END
/*
```

SYSIN DD

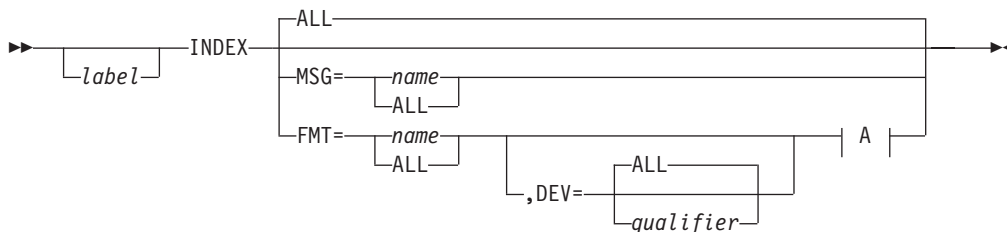
Defines the input control statement data sets.

Function Descriptions

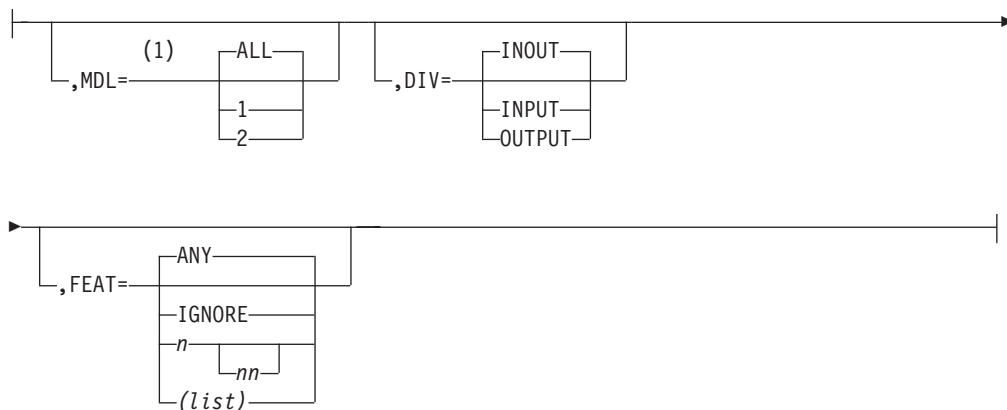
This section includes syntax diagrams and explanations for the INDEX, DELETE, SCRATCH, RELATE, and LIST functions.

INDEX

The INDEX function places the specified names of control blocks (or sets of related control blocks) in a special index directory (\$\$IMSDIR), that is used to provide quick online access to the control blocks.



A:



Notes:

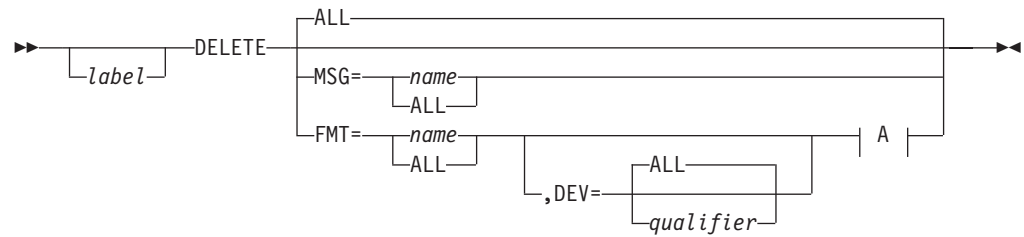
- 1 MDL=only applies to 3270 and 3270P devices.

If \$\$IMSDIR has been created by the INDEX function, it is read into the MFS buffer pool during initialization of the online IMS control region and made permanently resident there. If a requested control block is not found in the MFS buffer pool, the MFS pool buffer manager next looks for an entry in \$\$IMSDIR. The entry, if found, allows the MFS pool manager to issue a direct read for the control block.

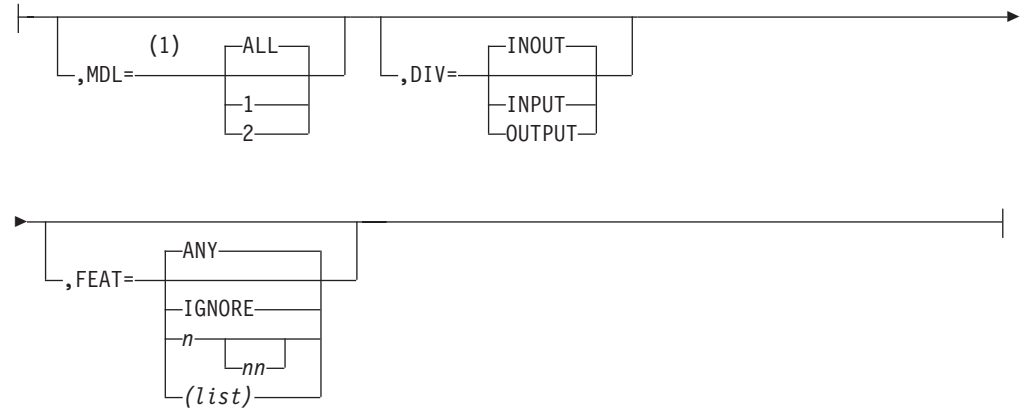
Using the special index directory to replace two direct-access storage reads with one is a performance advantage, but this advantage must be weighed against the storage cost in the online control region (14 bytes per control block indexed in the MFS buffer pool). Indexing only the most frequently used control blocks, which might be a small percentage of the total, can be advisable.

DELETE

The DELETE function specifies the names of a control block, or a set of control blocks, which are to be deleted from the special index directory (\$\$IMSDIR) used by the online MFS pool manager.



A:

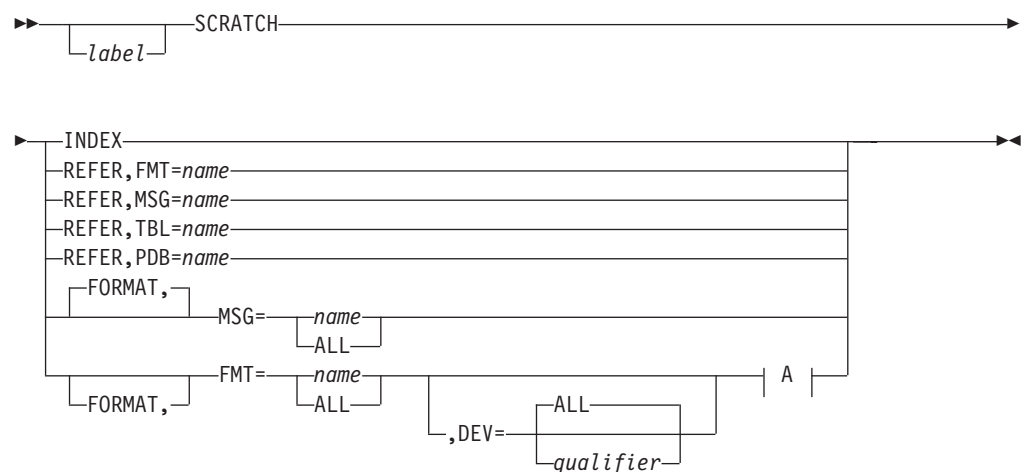


Notes:

- 1 MDL=only applies to 3270 and 3270P devices.

SCRATCH

The SCRATCH function scratches a message, format, partition set, or table ITB from IMS.REFERAL. Using this function, you can also scratch a message descriptor, device format, or an index directory from IMS.FORMAT or IMS.TFORMAT.



Total elimination of a control block requires the removal of its ITB as well. The 3270-1 control blocks for DFSDF2 could be deleted by recompiling the DFSDF2 format definition without the 3270-1 source included.

Another method is to use the MFS Service utility and scratch all of the DFSDF2 ITBs from the referral data set.

You can scratch the index directory using either DELETE ALL or SCRATCH INDEX.

Exception: The SCRATCH function does not apply to the MFS device characteristics table.

RELATE

The RELATE function provides a listing of all FMT, MSG, PDB, and TABLE ITBs in the IMS.REFERAL library.



FMT and MSG ITBs are related. Following each FMT ITB name, and indented three spaces, are the names of the MSG ITBs which include a SOR= parameter referencing that FMT. The MSG ITB entries indicate whether the message is INPUT or OUTPUT. A FMT ITB entry that does not exist in IMS.REFERAL, but is referred to by one or more MSG ITBs, is designated as **NOT DEFINED to the right of the entry. The PDB and TABLE ITBs are listed following the FMT and MSG ITBs.

Sample Output

Figure 14 shows the contents of IMS.REFERAL and the relationships between the FMT and MSG ITBs. For example, the first entry shows DFSDF1 as a FMT name which has two MSGs associated with it: DFSMI1 (an input MSG) and DFSMO1 (an output MSG).

```

RELATE
DFSDF1
  DFSMI1      INPUT
  DFSMO1      OUTPUT
DFSDF2
  DFSDSP01    OUTPUT
  DFSMI2      INPUT
  DFSMO2      OUTPUT

  DFSM03      OUTPUT
  DFSM05      OUTPUT
DFSDF3
  DFSMI4      INPUT
  DFSM04      OUTPUT
DFSDF5
  DFSMSTRI    INPUT
  DFS1209I    PROCESSING TERMINATED BY EOD ON SYSIN

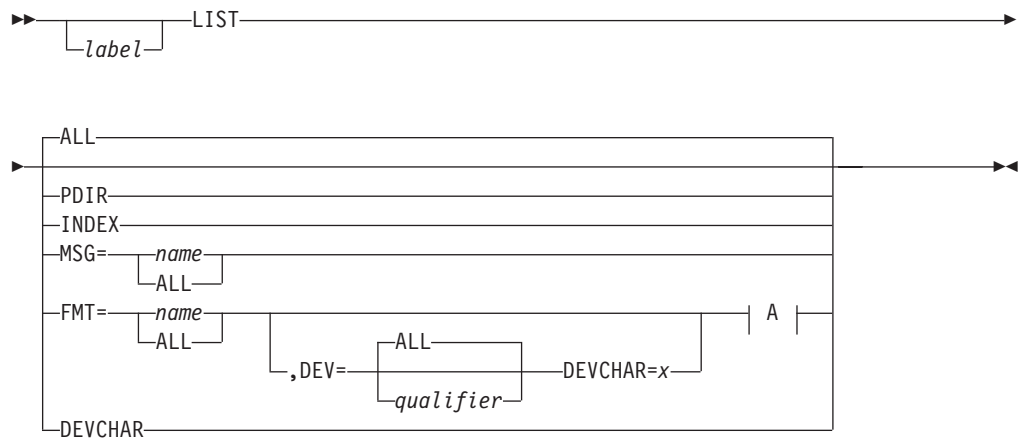
```

Figure 14. Example of a RELATE Output Listing

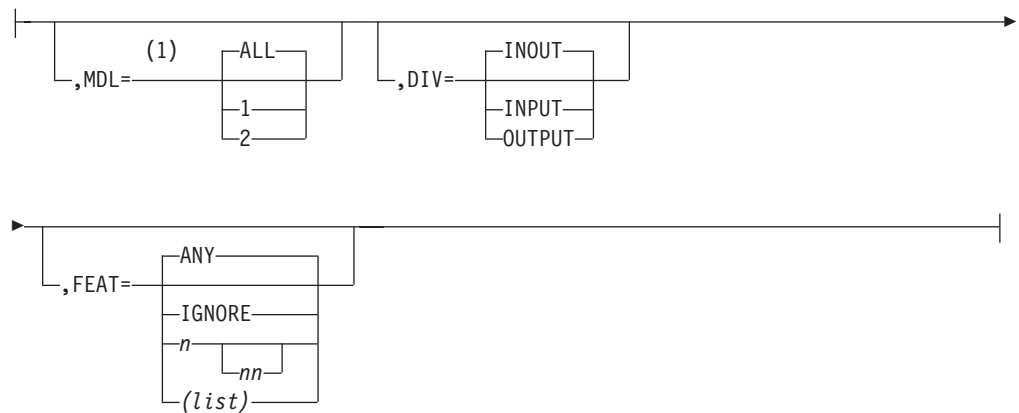
LIST

The LIST function provides an interpreted listing of contents of the test library IMS.TFORMAT, the staging library IMS.FORMAT, or the device characteristics table DFSUDT0X.

MFS Service



A:



Notes:

- 1 MDL=only applies to 3270 and 3270P devices.

For the staging library and the test library you can select the contents to be listed by qualifying the LIST control statement as follows:

ALL

Specifies both the PDS directory and the index directory. The default is ALL.

PDIR

Specifies the PDS directory.

INDEX

Specifies the index directory, \$IMSDIR.

MSG

Specifies the PDS directory entries for one or all message descriptors.

FMT

Specifies the PDS directory entries for one or more device formats.

A special case arises when the following chain of events occurs:

1. The Service Utility is started when the Index Directory (\$IMSDIR) does not exist.

2. The Index Directory is then created by the INDEX function.
3. LIST PDIR or LIST ALL is invoked.

In this case, the output containing the contents of the PDS directory does not include an entry for the Index Directory. This case occurs because the Index Directory is maintained in storage and is not written to the format library until the Service Utility ends.

The output listing contains a line for each control block with the following fields:

NAME

Represents the name of control block specified in the MSG or FMT statement.

TYPE

Represents the type of control block:

DIF

Device input format

DOF

Device output format

MSG

Message input or output control block

DEV

For FMT control blocks, represents the device type to which the control block applies.

MDL

For FMT control blocks, represents the model of the device type to which the control block applies. This field is used with 3270 and 3270P devices.

TTR

Represents the location of the control block in IMS.FORMAT (hexadecimal). Valid only for entries from IMS.FORMAT. Not valid for LIST INDEX.

SIZE

Represents the size of the control block in bytes (hexadecimal).

FEAT

For FMT control blocks, represents the hexadecimal uninterpreted representation of the device type, model, and specific features for which the control block applies. Because this information is appended to the FMT name in the library, this number determines the collating sequence of the output listing.

FEATURES

For FMT control blocks, represents the specific device features for which the control block applies.

Sample Output

Figure 15 on page 40 shows an example of the LIST function output listing for the control blocks described in Figure 13 on page 36.

MFS Service

NAME	TYPE	DEV	MDL	TTR	SIZE	FEAT	INTERPRETED FEATURES
DFSDF2	DIF	3270	1	000705	000001AE	007F	IGNORE
DFSDF2	D0F	3270	1	000703	00000279	007F	IGNORE
DFSDF2	D0F	3270P	1	00070B	000001A5	017F	IGNORE
DFSDF2	DIF	3270	2	000709	000000B6	027F	IGNORE
DFSDF2	D0F	3270	2	000707	00000296	027F	IGNORE
DFSDF2	D0F	3270P	2	00070D	00000195	037F	IGNORE
DFSMI2	MSG			00070F	0000005E		
DFSM02	MSG			000711	00000006		
DFSM03	MSG			000713	000000F3		

Figure 15. Example of a LIST Function Output Listing

LIST the MFS Device Characteristics Table

For the device characteristics table DFSUDT0x created during IMS system definition, the LIST DEVCHAR or DEVCHAR=0 or x, where x is the table suffix, provides an interpreted listing of all the entries of the MFS device characteristics table indicated. The listing shows the device symbolic name of the 3270 or SLU 2 devices, screen size, and features for each entry in the table. If the suffix is not specified in either the DEVCHAR parameter on the LIST statement or the PARM operand of the EXEC statement, table DFSUDT00 is listed. If the suffix is not specified in the DEVCHAR parameter or the LIST statement, but is specified in the EXEC statement, table DFSUDT0x (where x is the table suffix specified in the EXEC statement) is listed. The following examples are provided for the LIST DEVCHAR function:

Example 1

```
//MFSRVC EXEC PGM=DFSUTSA0, PARM='DEVCHAR=3'
//SYSIN DD *
LIST DEVCHAR
LIST DEVCHAR=7
```

Figure 16. Example of a LIST DEVCHAR Function with DEVCHAR=

The first LIST statement lists the contents of the device characteristics table DFSUDT03.

The second LIST statement lists the contents of the device characteristics table DFSUDT07.

Example 2

```
//MFSRVC EXEC PGM=DFSUTSA0
//SYSIN DD *
LIST DEVCHAR
LIST DEVCHAR=7
```

Figure 17. Example of a LIST DEVCHAR Function

The first LIST statement lists the contents of the device characteristics table, DFSUDT00.

The second LIST statement lists the contents of the device characteristics table, DFSUDT07.

LIST DEVCHAR Output

The output listing contains a line for each entry in the specified device characteristics table, which provides the following fields:

Symbolic Name

Symbolic name defined for the 3270 display in the TYPE= operand of the IMS system definition TYPE or TERMINAL macro.

Screen Size

Screen lines and columns of the 3270 display defined in the SIZE= operand of the IMS system definition TYPE or TERMINAL macro.

Device Features

Features specified in the FEAT= operand of the IMS system definition TYPE or TERMINAL macro.

Sample output

Figure 18 is an example of a LIST DEVCHAR output.

SYMBOLIC NAME	SCREEN LINES	SIZE COLS	DEVICE FEATURES
3270-A01	12	80	CARD, PFK, PEN
3270-A02	24	80	IGNORE
3270-A03	32	80	CARD, DEKYBD, PEN

Figure 18. Example of a LIST DEVCHAR Output

Utility Control Statements

The control statements used by the MFS Service utility program utilize the same syntax and many of the same keywords as the source statement input to the preprocessor.

Exception: An exception to this occurs when comments are placed on a statement or portion thereof.

Requirement: Because of the extreme range of allowable operations with a FMT block or blocks, the utility requires that comments on a statement be started with /* (for example, LIST ALL /* THIS STATEMENT...).

Related Reading: For a discussion of the meaning and use of keywords, see *IMS/ESA Application Programming: Transaction Manager*.

Positional Parameters

ALL

Where allowed, implies invocation of all functions supported for the associated operator. For example, INDEX ALL implies the insertion of all MID, MOD, DIF, and DOF names that exist in IMS.FORMAT into the special index directory (\$\$IMSDIR) to be used by the online control region for direct block access.

PDIR

Implies invocation of the associated operation against the PDS directory entries of IMS.FORMAT or, when used with LIST or SCRATCH, the PDS directories of IMS.TFORMAT. For example, LIST PDIR causes the contents of IMS.FORMAT or IMS.TFORMAT to be listed in interpreted format.

INDEX

Directs the invocation of an operation to the special index directory (\$\$IMSDIR) used by the online control region.

REFER

Directs the invocation of an operation to IMS.REFERAL used by the MFS language utility as an historical intermediate text storage library.

FORMAT

Directs the invocation of an operation to IMS.FORMAT, or, when used with LIST or SCRATCH, to IMS.TFORMAT.

DEVCHAR

Valid only with the LIST function. It causes the device characteristics table identified by the suffix (DEVCHAR=suffix) in the EXEC parameter to be printed. If the EXEC parameter is not specified, the contents of DFSUDT00 are printed.

Keywords

MSG=*name* | ALL

Directs the invocation of a function

name Directs the invocation of a function to a specific message control block, MID or MOD. Name must be specified as a 1- to 8-character alphanumeric value, the first character of which must be alphabetic.

ALL Directs the invocation of an operation to all message descriptors.

FMT=*name* | ALL

Directs the invocation of an operation

name Directs the invocation of an operation to a specific device format.

ALL Directs the invocation of an operation to all device formats. Unless further qualified, the operation proceeds against all FMT control blocks in IMS.FORMAT.

The FMT control block can consist of multiple FMT control blocks (DIFs and the DOFs) in IMS.FORMAT and, unless further qualified, the operation proceeds against all FMT control blocks with the same root name. You can specify "name" as a 1- to 6-character alphanumeric value, the first character of which must be alphabetic.

TBL=*name*

Directs the SCRATCH function to scratch a TBL ITB from the IMS.REFERAL library. The keyword is valid only on the SCRATCH utility control statement and only if the REFER positional parameter is specified. The names of all the TBLs that reside in IMS.REFERAL can be obtained through the Service Utility RELATE function.

PDB=*name*

Directs the function to scratch a PDB ITB from the IMS.REFERAL library. The keyword is valid only on the SCRATCH utility control statement and only if the REFER positional parameter is specified. The names of all the PDBs that reside in IMS.REFERAL can be obtained through the Service Utility RELATE function.

DEVCHAR=*x*

Causes the device characteristics table identified by the suffix following the "=" to be printed. This parameter is only valid with the LIST function.

DEV=

Qualifies a specified FMT control block as applying either to a particular device, a secondary logic unit type, or a remote program or to all (ALL).

Specify	Device
3270	3270 or SLU 2 display station
3270-A	For all 3270 or SLU 2 display stations that have been defined during IMS system definition using the device symbolic name.
3270-A1,...,A15	For a single 3270 or SLU 2 display station that has been defined during IMS system definition using the specific device symbolic name.
3270P	3270 printer
274X	2740/2741 terminals
FIN	Finance application program
FIDS	Finance display component (6x40) (for example, 3604-1 or 3604-2)
FIDS3	Finance display component (12x40) (for example, 3604-3)
FIDS4	Finance display component (16x64) (for example, 3604-4)
FIDS7	Finance display component (24x80) (for example, 3604-7)
FIJP	Finance journal printer
FIPB	Finance passbook printer
FIFP	Finance administrative printer
DPM-A	For all SLU P devices that have been defined during system definition using this device symbolic name.
DPM-B	For all ISC nodes that have been defined during system definition using this device symbolic name.
DPM-A1,...,A15	SLU P
DPM-B1,...,B15	ISC nodes
SCS1	The following console keyboard/printers: 3767; NTO; 3771; 3773; 3774; 3775; 3776; 3777; and SLU 1 (print data set) or SLU 4.
SCS2	3521 card punch, 3501 card reader, 2502 card reader; and SLU 1 (transmit data set) or, SLU 4
ALL	All devices listed above

MDL=1 | 2 | ALL

Determines FMT control block operation.

- 1 Restricts a FMT control block operation to control blocks for Model 1 3270/3270P stations.

- 2** Restricts a FMT control block operation to control blocks for Model 2 3270/3270P stations.

ALL

Directs FMT control block operation to control blocks for both Model 1 and Model 2 3270/3270P stations.

This keyword applies only to 3270 and 3270P devices.

DIV=INPUT | OUTPUT | INOUT

Determines FMT control block operation.

INPUT

Restricts a FMT control block operation to DIFs.

OUTPUT

Restricts a FMT control block operation to DOFs.

INOUT

Indicates a FMT control block operation is to proceed for both DIFs and DOFs.

FEAT=IGNORE | n[nn] | (list) | ANY

Determines FMT control block operation.

IGNORE

Restricts a FMT control block operation to control blocks for which FEAT=IGNORE was specified on the DEV statement to the MFS language utility.

n[nn]

With either:

- A print line of 120, 126, or 132
- User-defined features 1—10

(list)

Restricts a FMT control block operation to control blocks with a specific feature combination. The specifications allowed for “list” are indicated below.

For DEV=FIFP:

DUAL
132
(DUAL,132)

For DEV=3270 or DEV=3270-An:

PEN	,PFK	,CARD
<u>NOPEN</u>	DEKYBD	<u>NOCD</u>
	<u>NOPFK</u>	

Enter commas only where required to separate specifications that are actually coded. Feature specifications do not depend on position. You must code at least one alternative. The same feature value results, whether one, two, three, or none of the NOPEN, NOPFK, or NOCD parameters are specified.

ANY

Directs FMT control block operation to all control blocks without restrictions on the feature specification.

Chapter 4. Multiple Systems Verification Utility (DFSUMSV0)

The Multiple Systems Verification utility (DFSUMSV0) verifies the consistency and compatibility of system definitions for IMS systems in a multisystem environment. Use this utility with IMS Multiple Systems Coupling (MSC) when MTM, CTC, or VTAM is used. The use of this utility points out errors that can prevent the IMS systems from performing properly. If you do not use this utility, you must manually verify the compatibility of system definitions.

Recommendation: The Multiple Systems Verification utility should be run before attempting online executions to verify all defined links and routing paths.

If the MSVERIFY parameter is specified with the SYSTEM keyword on the IMSCTRL macro, only the IMS multisystem control block and verification utility are generated.

Related Reading: See the IMSCTRL macro in *IMS/ESA Installation Volume 2: System Definition and Tailoring* for the syntax to specify the MSVERIFY parameter.

In this Chapter:

- “Restrictions”
- “Prerequisites” on page 46
- “IMSMSV Procedure” on page 46
- “Input” on page 48
- “Output Messages and Path Map” on page 50
- “Utility Control Statements” on page 52
- “Error Processing” on page 53

Restrictions

The following restrictions apply to this utility:

- It cannot detect errors associated with Intersystem Communication (ISC). Refer to *IMS/ESA Administration Guide: System* for information on ISC.
- It does not support CICS.
- It cannot detect errors associated with directed routing.
- It cannot verify compatibility of the log write-ahead option between systems, because the option is specified when IMS is started. From 2 to 255 IMS systems can be verified in any one execution of the verification utility.
- It cannot be used to verify IMS subsystems within a shared queues group. MSC links between IMS subsystems in the same shared queues group are not supported.
- If the remote logical terminals (LTERMs) are incorrectly defined, the utility recognizes and issues error messages only for the remote LTERMs, leaving the local LTERMs in the target area undetected.
- Only MSC descriptors associated with the MSC links defined within the IMS system being initialized are processed by IMS. All other MSC descriptors are ignored.
- When verifying IMS systems with different release levels, use the utility from the latest release level.

MS Verification Utility

Before executing this utility, you must resolve all IMS definition errors in all the multisystem control blocks to be included in the total multisystem configuration. After all system definitions are complete, you must link-edit all the multisystem control blocks from all the IMS multisystem definitions into IMS.RESLIB or some other user-specified library. The utility then has access to the multisystem control blocks. For problem determination, assembly listings of all the IMS multisystem control blocks should be available when the utility is executed. Without the assembly listings, it will be difficult for you to resolve inconsistencies and incompatibilities displayed as a result of the multisystem verification process.

Prerequisites

This section describes the prerequisites for using the Multiple Systems Verification utility.

Before the verification utility can be executed, the multisystem control block modules for all systems to be verified must be loaded into IMS.RESLIB or some other user-specified library. Figure 19 provides a sample job stream that can load the multisystem control block modules into IMS.RESLIB:

```
//MSADD    JOB  (JOB STATEMENT PARAMETERS)
//*
//STEP1    EXEC PGM=IEWL
              PARM='SIZE=(200K),NCAL,LET,REUS,XREF,LIST'
//SYSPRINT DD SYSOUT=A
//SYSLMOD  DD DSN=IMS.RESLIB,DISP=OLD
//SYSUT1   DD UNIT=SYSDA,SPACE=(1024,(100,10))
//SYSLIN   DD *
              PLACE OBJECT MODULES HERE
              NAME DFSMSxxx(R)
/*
```

Figure 19. Sample Job Stream

You must provide two or more object modules. A NAME statement identifying the preceding multisystem control block module (replace xxx with the 3-digit control block name suffix) must follow each object deck. If a library other than IMS.RESLIB is used, modify the //SYSLMOD statement accordingly.

IMSMSV Procedure

This section describes the JCL for the IMSMSV procedure, the PROC statement, the EXEC statement, and invoking the procedure.

Figure 20 shows an example of the procedure used to execute the Multiple System Verification utility. IMSMSV is created at system definition and is placed in the IMS.PROCLIB by Stage 2 of SYSDEF.

```
//          PROC    DSN='IMS.RESLIB',
//                  REG=32K,CLASS=A,ALL=ALL,
//                  UNIT=SYSDA,SER=,DSM='IMS.MODBLKS'
//MSVERIFY EXEC    PGM=DFSUMSV0,PARM='&ALL',REGION=&REG
//STEPLIB  DD      DSN=&DSM,DISP=SHR,UNIT=&UNIT,VOL=SER=&SER
//          DD      DSN=&DSN,DISP=SHR,UNIT=&UNIT,VOL=SER=&SER
//SYSOUT   DD      SYSOUT=&CLASS
```

Figure 20. Multiple System Verification Utility Procedure

PROC Statement

The procedure statement must be in the form:

```
PROC    DSN='IMS.RESLIB',
        REG=32K,CLASS=A,ALL=ALL,
        UNIT=SYSDA,SER=,DSM='IMS.MODBLKS'
```

ALL=

Specifies that all messages, including the information message DFS2327I, are to be printed. The default is ALL.

CLASS=

Specifies the SYSOUT class. The default is A.

DSM=

Specifies the data set name of IMS.MODBLKS.

DSN=

Specifies the data set name that contains the verification utility program (DFSMSV00) and its control blocks. The default is IMS.RESLIB.

REG=

Specifies the region size for this execution. The default is 32KB.

SER=

Specifies the volume serial number of the DASD that contains the data set specified by the DSN parameter. SER= need not be specified if the data set is a cataloged data set.

UNIT=

Specifies the STEPLIB UNIT TYPE.

EXEC Statement

The execution statement must be in the form of PGM=DFSUMSV0. The PARM= field must be in the form:

```
PARM='&ALL',REGION=&REG
```

If PARM=ALL is specified in the EXEC statement for the utility execution, the information message DFS2327I is printed as part of the utility output. This message warns you not to do an //ASSIGN of the SYSID/MSNAME to the logical link referenced, because the assignment does not provide a path to the local SYSID at this SYSID level.

Invoking the Procedure

The JCL required to execute the verification utility is shown in Figure 21.

```
//MSVERIFY JOB (JOB STATEMENT PARAMETERS)
//STEP1    EXEC IMSMSV
//SYSIN     DD *
            INPUT FOR IMS MULTISYSTEM VERIFICATION UTILITY
/*
```

Figure 21. JCL Requirements for the MS Verification Utility

Input

This section describes the required input for the Multiple Systems Verification utility.

This utility processes input in two phases:

- Input validation
- Multisystem control block verification

Input Validation

Requirement: The verification utility requires as input one or more control statements within a SYSIN data set. Refer to “Utility Control Statements” on page 52.

After the input has been validated, this phase prints a list of the valid multisystem control block names. The utility then determines if the multisystem control block names are in the IMS.RESLIB PDS directory. The utility prints any control block names not found in the directory. If any errors have been detected up to this point, the utility terminates execution with a completion code of 12. If no errors have been found, the utility loads the multisystem control blocks into real storage.

Multisystem Control Block Verification

The utility verifies the following specific portions of each multisystem control block:

- Partner IDs and assigned physical links
- Remote SYSID to Local SYSID Paths
- Remote and local transaction attributes
- Presence of corresponding logical terminals

Partner IDs and Assigned Physical Links

The partner IDs in the logical link definitions are verified to ensure that a partner ID is:

- Not referenced in only one system
- Referenced in only two systems

Each partner ID, as defined with the PARTNER keyword on the MSLINK macro, is checked against every other partner ID in every other multisystem control block. Appropriate messages are printed if any errors are found. Logical links in error are treated as undefined in subsequent steps of the verification process.

When a partner ID is verified and there is also an MSPLINK (physical link) defined for this logical link in both systems, the physical link attributes are verified for type and buffer size. The following are physical link types:

- Real storage-to-real storage
- Channel-to-channel
- VTAM

If any physical link incompatibility is found, the attributes of both physical link definitions are displayed to assist you in determining the error. If the MSPLINK is defined for only one logical link, an information message is printed, indicating that the other end is undefined. In addition to the MSC physical and logical links that are defined to IMS through system definitions, you can identify remote names to IMS through an MSC descriptor. The MSC descriptor relates each remote resource with the link name of a generated MSNAME macro.

Remote SYSID to Local SYSID Paths

The SYSID table entries for a SYSID are verified across all multisystem control blocks for that SYSID number to determine if any path errors exist. A path error is an incomplete path between a multisystem control block in which the SYSID is defined as remote and the multisystem control block in which the SYSID is defined as local.

An incomplete path can occur for the following reasons:

- A SYSID in a multisystem control block for an intermediate system does not contain an address of an MSNAME block (undefined SYSID).
- The MSNAME block is assigned to a logical link block that has a path back to itself without a local SYSID (loop condition).
- The MSNAME block is associated with a logical link block that has an invalid partner ID. The partner ID is invalid if it is not defined in any other multisystem control block or if it is defined in more than one other multisystem control block.
- A SYSID number is defined as local in more than one system.
- A SYSID number is not defined as local in any system.

SYSIDs are scanned in numeric order until all logical link paths are verified. After a path is verified, the utility might display warning messages. These messages identify which logical link numbers this SYSID number-MSNAME should not be assigned to, because an invalid path to the local SYSID would result.

Remote and Local Transaction Attributes

Each multisystem control block is scanned for remote transaction definitions. Each remote transaction definition references a remote and local SYSID. Each remote transaction code is compared with the transaction codes in the system where the remote SYSID is defined as local. If no matching transaction code is found, an error message is printed. If a match is found, the attributes are verified in the two multisystem control blocks.

The following transaction code attributes must be consistent between systems:

- Local
- Remote
- Recoverable
- Nonrecoverable
- Conversational
- Fixed scratch pad area
- Fixed scratch pad area length
- Non-inquiry
- Inquiry
- Single segment
- Multisegment
- Non-Fast Path

If a discrepancy in the transaction attributes occurs, an error message is printed. The attributes specified for the transaction in both systems are displayed.

The return from the local transaction is checked to ensure a path exists out of the local system back to the corresponding remote transaction. If an error exists, a message is printed.

MS Verification Utility

Presence of Corresponding Logical Terminals

Each multisystem control block is scanned for remote LTERM definitions. Each LTERM is associated with an MSNAME block. Each MSNAME block contains remote and local SYSID definitions.

When a remote LTERM is found, the utility checks to ensure that a corresponding LTERM is defined with the same name in the system where the remote SYSID for the MSNAME is defined as local. If no corresponding LTERM definition is found, an error message is printed. If an LTERM is found, verification continues.

For multisystem LTERMs defined during IMS system definition, the remote LTERM definition can be made through the IMS system definition for the remote system or by the Extended Terminal Option (ETO) MSC descriptors for the remote system. When you use ETO MSC descriptors, the remote LTERMs do not exist until the initialization of the ETO feature on the remote IMS system. Therefore, an error message is printed by this utility for the missing remote LTERM indicating that the LTERM might be a dynamic resource.

The utility checks the return path from the destination LTERM to ensure that a path exists out of the local system back to the corresponding remote LTERM. If an error exists, a message is printed.

The return path to that system is the SYSID defined as local in the MSNAME block in the multisystem control block in which the LTERM was defined as remote.

After all verification work is complete, the utility prints a path map as an aid in visualizing the configuration of systems.

Output Messages and Path Map

This section describes the utility output and the path map. The verification utility output includes information, warning, and error messages and a path map.

Related Reading: The messages are defined in *IMS/ESA Messages and Codes*.

Because an error can cause other error conditions, messages with lower numbers should be analyzed and corrected first. The path map is a summary, in matrix format, of the routing paths verified by the utility. It is produced for the first 18 or fewer systems (in ascending sequence by multisystem control block name) being verified. If more than 18 systems are being verified, verification for all systems occurs, appropriate messages for all systems are provided, but the path map is provided only for the first 18 systems.

Figure 22 on page 51 is a sample path map for a configuration of five systems with multisystem control block names of DFSMS002, DFSMS003, DFSMS004, DFSMS005, and DFSMS006. (For the sake of simplicity, these names are used to refer to the specific systems in the rest of this discussion.)

A path map has two sections. The top section relates SYSIDs to specific systems. The bottom section relates partner IDs to logical links between specific systems. The contents of the path map are keyed by letters that are defined after Figure 22 on page 51.

A ↘

	MS002	MS003	MS004	MS005	MS006
* 001	0006 AB ← C	*** BD	0002 BD	0006 AE	LOCAL
* 002	*** BC	0006 AC	0002 BD	0006 AE	
* 003	0004 BD	*** BC	0006 DA	0006 AE	LOCAL
004	LOCAL	0006 AC	0006 DA	0006 AE	0002 AB
* 005	LOCAL	*** BC	0006 DA	0006 AE	0003 AC
* 006	0006 AB	0006 AC	0002 BD	0006 AE	LOCAL
007	*** BC	LOCAL	0006 DA	0006 AE	0003 AC
* 008	*** BC	LOCAL	0006 DA	0006 AE	0002 AB
* 009	*** BC	LOCAL	0002 BD	0006 AE	0004 AD
* 010	0006 AB	*** BC	LOCAL	0006 AE	0004 AD
* 011	*** BC	0006 AC	LOCAL	0006 AE	0003 AC
* 012	0004 BD	*** BC	LOCAL	0006 AE	0002 AB
013	0004 BD	0006 AC	LOCAL	0006 AE	0002 AB
* 014	0006 AB	*** BC	LOCAL	0006 AE	0003 AC
* 015	*** BC	0006 AC	0002 BD	LOCAL	** AX
* 016	0004 BD	*** BC	0002 BD	LOCAL	0005 AE
* 017			0006 DA	LOCAL	
018					
019	← B				LOCAL
020					
021					
022					
023					
024					
025					
026					
027					
028					
029				0002 AB	
* 030					
VTM	E → AB 0001				AB 0001
VTM	BC 0002	BC 0002			BC 0005
VTM	BD 0003		BD 0002		
VTM		AC 0001			AC 0002
VTM	← D		AD 0001		AD 0003
VTM			DA 0003		DA 0005
MTM				AE 0001	AE 0004
VTM					AX 0005
VTM					AF 0006

* AN ERROR WAS DETECTED ON THIS LINE
 ** NO PARTNER FOR ID
 *** MULTI-PARTNERS FOR ID
 DFS2399I JOB TERMINATED - RETURN CODE12

Figure 22. Sample Multisystem Path Map

Notes to Figure 22:**Letter Meaning**

- A** The top row contains the multisystem control block names (not including the DFS prefix) of the systems verified by execution of this utility.
- B** The first column of the top section contains all SYSIDs defined in the multisystem configuration. An asterisk preceding the SYSID number indicates an error exists on this line of the matrix.
- C** Each entry in the top section relates the SYSID (column B) to the multisystem control block name (row A).
- Most entries contain the 4-digit suffix of the multisystem control block name of the system defined as logically linked to the system (row A) and the 2-character partner ID defined for this logical link.
 - A blank entry, such as SYSID 0002 for DFSMS006, indicates this SYSID was not defined for this system. All entries for SYSIDs

MS Verification Utility

0020 through 0029 are blank; this means these SYSIDs were not specified in any of the multisystem control blocks.

- An entry specifying LOCAL, such as SYSID 0004 for DFSMS002, identifies this system as the system in which this SYSID is defined as local.
- Errors are identified by asterisks. One asterisk preceding the SYSID indicates that one or more errors were found for this printed line. If the suffix portion of an entry is replaced by two asterisks (**), such as SYSID 0015 for DFSMS006, the verification utility found no partner for this system. Three asterisks (***) indicate that more than two partners were found, such as SYSID 0002 for DFSMS002. If a SYSID is defined as local for more than one system, the printed line is identified by one asterisk and more than one entry on that line specifies LOCAL. SYSID 0006 contains an example of this error.

D The first column of the bottom section contains the physical link type:

CTC	Channel-to-channel adapter
MTM	Real storage-to-real storage
VTM	ACF/VTAM session type 6

This column entry is either blank, if no physical link is defined, or assigned depending on the first physical link encountered, for the logical link identified by E. An asterisk preceding the link type (or alone, if no physical link is defined) indicates an error exists for this printed line.

E Identifies the logical link in terms of partner ID and relative logical link number. The partner IDs relate directly to those in the top part of the chart.

The verification utility relates partner systems by connecting them with a dashed line.

In Figure 22 on page 51, partnerships BC and AX are in error. These errors were identified in the top part of the chart but are more clearly demonstrated in the bottom part. Partner ID BC has more than two definitions; AX has just one definition.

Utility Control Statements

This section describes and provides examples of the statement input.

The control statements must contain the 1- to 3-digit suffix supplied on the MSVID keyword of the IMSCTRL macro. Each control statement can contain one or more such suffixes, specified in any sequence. The input statement scan ends when a blank position is encountered; if position 1 is blank, the input statement is treated as a comment statement. If more than one suffix is specified in a control statement, each suffix following the first one must be separated from the preceding suffix by a comma. Only the significant digits of a suffix need be specified.

Each suffix in a control statement must be complete in that statement and cannot be continued in the next control statement.

Sample input data:

- 1,255,6,009,80,02,198 are valid.
- 0,677,040,NYC,1A,0001,5,5 are invalid.

Each invalid entry is printed, and the type of error is identified.

For example:

0	Is not in the range from 1 to 676.
677	Is not in the range from 1 to 676.
0040	Is more than 3 digits.
NYC	Is nonnumeric.
5,5	Is duplicate input data.

Valid input for three systems:

STATEMENT	1,5,255
or	
STATEMENT	001,005,255
or	
STATEMENT	255,01,5
or	
STATEMENT	1 DFSMS001 NYC
and	
STATEMENT	255 DFSMS255 LA
and	
STATEMENT	05 DFSMS005 CHICAGO

Error Processing

This section explains the condition codes returned by the utility.

These condition codes are:

Code	Meaning
0	Only information and warning messages are printed
12	Errors detected that must be resolved before multisystem execution

Chapter 5. Spool SYSOUT Print Utility (DFSUPRT0)

When a communication line is defined for Spool SYSOUT during system definition, the Spool SYSOUT Print Utility (DFSUPRT0) copies messages produced by the online control program from its set of data sets to a system output device. Both the spool data sets and the system output device are processed using QSAM. Blocking factors for spool data sets are determined by the online control program. System output device blocking can be specified through JCL on the SYSPRINT DD statement.

In this Chapter:

- “Restrictions”
- “Input and Output”
- “DFSWTnnn Procedure”
- “IMSWTnnn Job” on page 57
- “Error Processing” on page 57

Restrictions

The Spool SYSOUT Print utility does not support CICS.

Input and Output

This section describes the input and output for the utility.

When the print utility is started, it should complete before a /START LINE command is issued to make the Spool SYSOUT available.

Output from the print utility includes a page of status information, followed by the contents of the spool data sets indicated as FULL and printed in chronological sequence.

DFSWTnnn Procedure

This section explains the JCL for the DFSWTnnn procedure, the PROC statement, the EXEC statement, and the DD statements.

The DFSWTnnn procedure, shown in Figure 23 on page 56, executes the Spool SYSOUT Print utility program (DFSUPRT0) as an online program for printing data sets created by the Spool SYSOUT option during system definition. The utility program copies messages produced by the online control program from its set of data sets to a system output device. This procedure is created at system definition and is placed in the IMS.PROCLIB procedure library by Stage 2 of SYSDEF. The *nnn* is supplied by system definition.

Spool SYSOUT Print

```
//      PROC   SOUT=A,RGN=30K,SYS1=,SYS2=
//PRINT EXEC  PGM=DFSUPRT0,REGION=&RGN
//STEPLIB DD   DSN=IMS.&SYS2.RESLIB,DISP=SHR
//SYSPRINT DD  SYSOUT=&SOUT,DCB=BLKSIZE=1410;
//SYSUDUMP DD  SYSOUT=&SOUT
//SPOOLn DD    DISP=SHR,DSN=IMS.&SYS1.SYS01
//SPOOLn DD    DISP=SHR,DSN=IMS.&SYS1.SYS02
//SPOOLn DD    DISP=SHR,DSN=IMS.&SYS1.SYS03
```

Figure 23. DFSWTnnn Procedure

PROC Statement

This statement must be in the form:

```
PROC   SOUT=A,RGN=30K,SYS1=,SYS2=
```

SOUT=

Specifies the class assigned to SYSOUT DD statements.

RGN=

Specifies the size of the MVS region to be allocated to the IMS control program.

SYS1=

Specifies an optional second level dsname qualifier for those data sets which are designated as “Mandatory Replicate” in an XRF complex. When specified, the operand must be enclosed in quotes and must include a trailing period; for example, SYS1='IMSA.'.

SYS2=

Specifies an optional second level dsname qualifier for those data sets which are designated as “Optional Replicate” in an XRF complex. When specified, the operand must be enclosed in quotes and must include a trailing period; for example, SYS2='IMSA.'.

EXEC Statement

This statement can be in the form:

```
EXEC   PGM=DFSUPRT0
```

or it can specify a procedure that contains the required JCL statement. A region size of 30KB is usually adequate for execution.

DD statements

STEPLIB DD

Defines the library containing the print utility. This DD statement is usually DSN=IMS.RESLIB,DISP=SHR.

SYSPRINT DD

Defines the system output device to which output is directed. The record format is VBM. If either no block size or a block size less than 141 is specified, the default block size of 141 is assumed. Any block size valid for QSAM and greater than 141 can be specified. Any logical record length can be specified. If no logical record length is specified, the default is 4 less than the block size specified (137 if block size default of 141 is used).

SPOOLnn DD

Describes the spool data set to be printed (where nn is any valid alphanumeric identifier). This DD statement is normally DSN=IMS.SYSnn, where 'nn' is assigned by system definition. DCB information either should not be coded or, if coded, must specify RECFM=VBM.

IMSWTnnn Job

This section explains the job used to print the data sets created by the utility.

To invoke the DFSWTnnn procedure, use a IMSWTnnn job. These jobs print data sets created by the Spool SYSOUT options.

IMSWTnnn member job class and message class are determined by the MAXREGN keyword specified on the IMSCTRL macro statement during system definition.

This job executes procedure DFSWTnnn, which invokes the Spool SYSOUT utility program (DFSUPRT0) for printing the Spool SYSOUT data set.

```
//SPRT0 JOB 1,IMS,CLASS=A,MSGCLASS=A,MSGLEVEL=1
//          EXEC DFSWT000
```

This job must be copied to the IMS.JOBS data set to run.

Sample Output

Figure 24 shows an example of the Spool SYSOUT Print utility output.

DFSUPRT0 - SYSOUT PRINT UTILITY			TIME	9:35:1	DATE	92.056
DDNAME	STATUS	CREATED TIME	DATE	DATASET NAME		
SP00L1	FULL	9:33:239	92.056	IMSTESTL.IMS01.SP00L1		
SP00L2	INUS	:00:000	0.000	IMSTESTL.IMS01.SP00L2		
SP00L3	AVAL	:00:000	0.000	IMSTESTL.IMS01.SP00L3		

Figure 24. Example of a Spool SYSOUT Print Utility Output

The fields in the report have the following meaning:

DDNAME The user-provided DDNAME

STATUS FULL—if data set is to be printed
 INUS—if being filled online
 AVAL—if not being used

CREATED TIME
 Time of creation (24-hour clock) (HH:MM:SST)

DATE Julian date of creation (YY.DDD)

DATASET NAME
 The DSNAME of the assigned data set

System messages included in a spool data set always have unprintable control characters (typically the new-line symbol, X'15'). If a UCS printer is used as a SYSOUT device, these messages might print as extraneous alphabetic characters (if fold-mode operation is specified in response to the UCS parameter request).

Error Processing

This section explains the condition codes returned by the utility.

These condition codes are:

Spool SYSOUT Print

Code	Meaning
0	Successful completion
4	No data sets allocated for printing
8	SYSPRINT DD statement missing
12	I/O error on SYSPRINT data set

Chapter 6. Time-Controlled Operations Verification Utility (DFSTVER0)

The Time-Controlled Operations (TCO) Verification utility (DFSTVER0) ensures error-free TCO script members. You should run the verification utility before you execute any script member online. The utility detects any script member error that would be detected during online execution.

Exception: Errors caused by insufficient storage are not detected.

You must add TCO script members to the TCO script library before executing the verification utility. You can verify more than one member at a time by assigning an input control statement for each member you are verifying.

The verification utility generates reports for the following:

- Errors
- Statistics
- Timer elements (time-schedule requests)
- Messages
- Summaries

Related Reading: For more information on time-controlled operations, see *IMS/ESA Operations Guide* .

In this Chapter:

- “TCO Verification Procedure”
- “Output” on page 61
- “Return Codes” on page 64

TCO Verification Procedure

This section explains the JCL for the utility, the EXEC statement, and the DD statements.

Figure 25 on page 60 is a sample script member (DFSTCF10). Figure 26 on page 60 shows the JCL for the TCO Verification utility procedure for Figure 25 on page 60.

TCO Verification

```
/BRO LTERM CTRL
DFSTCF10 LOADED.
*TIME      DFSTXITB          S          ****
/ASS LTERM LOG27403 TO LINE 31 PTERM 1 ;
/START LINE 2 PTERM ALL;
/START LINE 26 PTERM ALL;
/START LINE 18 PTERM ALL;
/STA DB MSDBLM01,MSDBLM02,MSDBLM03,MSDBLM04,MSDBLM05;
/STA DB MSDBLM06,MSDBLM07,MSDBLM08;
*TIME      DFSTXITB          S          ****
/START REGION MSDBMTX3;
/START REGION MSDBMTY3;
/START REGION MSDBMTZ1;
*TIME      DFSTXITB          S          ****
PTERM01 BEGIN PTERM1;
PTERM03 BEGIN PTERM3;
/STOP REGION 1;
*TIME      DFSTXITB          S          ****
DFSTCF LOAD DFSTCF1A;
*TIME      DFSTXITB          0004 S      ****
*TIME      DFSTXITB          0004 S      ****
/*
```

Figure 25. Sample Script Member (DFSTCF10)

The following example is for the verification of members DFSTCF01, DFSTCF02, and DFSTCF10.

```
//VERIFY JOB 1,MSGLEVEL=1
/*
/*      THIS JCL IS USED TO VERIFY THE USER SUPPLIED SCRIPTS
/*
// EXEC PGM=DFSTVER0
//STEP CAT DD DSN=VCATQAV,DISP=SHR
//STEPLIB DD DSN=IMS.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//DFSTCF DD DSN=IMS.TCFSLIB,DISP=SHR
//SYSIN DD *
DFSTCF01 CONT 5
DFSTCF02 CONT 20
DFSTCF10      (ONE OR MORE CARDS SPECIFYING MEMBER NAMES TO BE VERIFIED)
/*
```

Figure 26. TCO Verification Utility Procedure

In this case, the three script members DFSTCF01, DFSTCF02, and DFSTCF10 are in IMS.TCFSLIB, referred to in the DFSTCF DD statement. The verification utility, DFSTVER0, is found in IMS.RESLIB, referred to in the STEPLIB DD statement. The reports generated by the verification utility are sent to the device you assign for class A output.

EXEC Statement

The TCO Verification utility is executed as a standard MVS job. The following are required:

- A JOB statement defined by you
- An EXEC statement
- DD statements

The region size for execution of the utility is acquired in 4KB increments for time-schedule request sets and message sets. Each run of the utility, therefore, causes 8KB to be acquired. Each timing element uses 32 bytes of the 4KB of storage. The amount of storage messages use varies, depending on the number of segments you specify. In the example job, more time-schedule requests and messages could be added without increasing the storage.

EXEC

Must be in the form:

```
EXEC    PGM=DFSTVER0
```

DD Statements

STEPCAT DD

Describes a private VSAM user catalog that is searched first. This is required if the defined areas are cataloged in a user catalog.

STEPLIB DD

Points to IMS.RESLIB, where the TCO modules reside.

SYSPRINT DD

Describes the output data set that contains the reports the verification utility generates.

SYSUDUMP

Defines the dump data set.

SYSIN DD

Describes the input control data set, which contains the 80-character control statements. The TCO script member names you are verifying with the utility are in columns 1 through 8 of each statement.

The CONT keyword with its parameter can be used to change the size of a message segment; the default segment continuation count is 9. The CONT parameter is a 1- or 2-digit number between 1 and 99 that indicates the new segment continuation count for that particular script.

DFSTCF DD

Points to IMS.TCFSLIB, where the TCO script members reside. You can name the data set TCFSLIB or any other valid dsname.

Output

This section explains and gives examples of the reports generated by the TCO Verification utility.

Error Report

Figure 27 on page 62 is an example of an error report generated by the TCO Verification utility. In the report, the statement sequenced 00003400 (in the DFSTCF10 script member) had a misspelled exit routine name. This exit routine could not be found in the IMS.RESLIB library, so it is reported as an error here. The statement is eliminated from the time-schedule request table, which is in the timer elements report.

TCO Verification

```
ERROR REPORT FOR MEMBER DFSTCF10  
  
DFS3360E USER EXIT DFSTXTIB REQUESTED NOT FOUND, SEQUENCE NUMBER= 00003400
```

Figure 27. TCO-Verification-Error Report

Statistics Report

Figure 28 shows an example of a statistics report generated by the TCO Verification utility. The only exit routine specified by any time request statement (that was found in IMS.RESLIB) is DFSTXITB.

```
STATISTICS REPORT FOR MEMBER DFSTCF10  
  
PROGRAM EXITS REQUIRED IN IMS.RESLIB  
  
DFSTXITB
```

Figure 28. TCO-Verification-Statistics Report

Timer-Elements Report

Figure 29 shows an example of a timer-elements report generated by the TCO Verification utility.

```
TIMER ELEMENTS REPORT  
  
TIME OF ACTIVATION EXIT CALLED  ATTRIBUTES  PARM  
  
STARTUP          DFSTXITB  RES          **** MSG SET  
STARTUP          DFSTXITB  RES          **** MSG SET  
STARTUP          DFSTXITB  RES          **** MSG SET  
STARTUP          DFSTXITB  RES          **** MSG SET  
0957             DFSTXITB  RES SNGL     **** MSG SET
```

Figure 29. TCO-Verification-Time-Elements Report

The columns in the report are as follows:

Time of Activation

Indicates either STARTUP or the time the time request is to be processed if this is an actual run.

Exit Called

Indicates the exit to be called for this time request.

Attributes

The following attributes are possible:

RES

Indicates a resident exit routine.

DYN

Indicates a dynamically loaded exit routine.

CONT

Indicates execution each day at the same time.

SNGL

Indicates a single execution.

PARM

Indicates either ****MSG SET, which indicates a message set for this time request, or the 16 bytes of data specified in columns 56 through 71 for this time request in the script member.

Message-Table Report

Figure 30 is an example of a message-table report generated by the TCO-Verification utility. The report indicates that the DFSTCF10 script member has five message sets. The asterisk in column 1 indicates a new message. Each message set is separated by a blank line.

```
MESSAGE TABLE REPORT

EACH LINE IS A SEGMENT
* IN COLUMN 1 SIGNIFIES START OF NEW MESSAGE
* IN COLUMN 121 SIGNIFIES SEGMENT IS TRUNCATED

*/BRO LTERM CTRL
DFSTCF10 LOADED.

*/ASS LTERM LOG27403 TO LINE 31 PTERM 1 ;
*/START LINE 2 PTERM ALL;
*/START LINE 26 PTERM ALL;
*/START LINE 18 PTERM ALL;
*/STA DB MSDBLM01,MSDBLM02,MSDBLM03,MSDBLM04,MSDBLM05;
*/STA DB MSDBLM06,MSDBLM07,MSDBLM08;

*/START REGION MSDBMTX3;
*/START REGION MSDBMTY3;
*/START REGION MSDBMTZ1;

*PTERM01 BEGIN PTERM1;
*PTERM03 BEGIN PTERM3;
*/STOP REGION 1;

*DFSTCF LOAD DFSTCF1A;
```

Figure 30. Example of a Message-Table Report

A message set is composed of one or more messages. A message set is either single-segment or multi-segment. In the sample report:

- The first message set is a single multi-segment message.
- The second, third, and fourth message sets are multiple single-segment messages.
- The last message set is a single single-segment message.

Summary Report

Figure 31 on page 64 is an example of a summary report generated by the TCO-Verification utility. The summary report lists the number of time-schedule requests and the number of messages found in a script member. It also summarizes the number of exit routines specified in the time-schedule requests in the member being verified and the amount of storage used.

TCO Verification

SUMMARY REPORT

# ELEMENTS	# MSGS	#EXIT ROUTINES	STORAGE SIZE
00005	00014	00001	08324

Figure 31. TCO-Verification-Summary Report

Return Codes

This section explains the return codes for the TCO Verification utility.

The TCO Verification utility returns a code that indicates the verification processing status. These return codes are:

Code	Meaning
------	---------

- | | |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0 | No error found in the scripts being verified. The five output reports are generated for each script. |
| 4 | Error in the CONT parameter that was specified on the verification JCL. |
| 6 | Syntax errors in the script. |
| 8 | One of the following errors occurred: <ul style="list-style-type: none">• Unable to get storage• Unable to open SYSIN data set• Unable to open SYSPRINT data set• No DFSTCF DD statement in Verification JCL |
| 10 | Error in I/O. |
| 12 | Script member not found. |
| 14 | Unable to open the script members data set. |

If a return code greater than zero is received from the utility, one or more of the scripts being verified has errors.

Verification utility reports are issued for reports that have no errors. Only the error report is issued for scripts that have errors. However, depending on the type of error, it is possible that no error report is generated.

Part 3. Appendixes

IMS/ESA Version 6 Library

SC26-8725	ADB	Administration Guide: Database Manager
SC26-8730	AS	Administration Guide: System
SC26-8731	ATM	Administration Guide: Transaction Manager
SC26-8727	APDB	Application Programming: Database Manager
SC26-8728	APDG	Application Programming: Design Guide
SC26-8726	APCICS	Application Programming: EXEC DLI Commands for CICS and IMS
SC26-8729	APTM	Application Programming: Transaction Manager
SC26-8732	CG	Customization Guide
SC26-9517	CQS	Common Queue Server Reference
SC26-8733	DBRC	Database Recovery Control Guide and Reference
LY37-3731	DGR	Diagnosis Guide and Reference
LY37-3732	FAST	Failure Analysis Structure Tables (FAST) for Dump Analysis
GC26-8736	IIV	Installation Volume 1: Installation and Verification
GC26-8737	ISDT	Installation Volume 2: System Definition and Tailoring
SC26-8740	MIG	Master Index and Glossary
GC26-8739	MC	Messages and Codes
SC26-8743	OTMA	Open Transaction Manager Access Guide
SC26-8741	OG	Operations Guide
SC26-8742	OR	Operator's Reference
GC26-8744	RPG	Release Planning Guide
SC26-8767	SOP	Sample Operating Procedures
SC26-8769	URDB	Utilities Reference: Database Manager
SC26-8770	URS	Utilities Reference: System
SC26-8771	URTM	Utilities Reference: Transaction Manager

Supplementary Publications

GC26-8738	LPS	Licensed Program Specifications
SC26-8766	SOC	Summary of Operator Commands

Online Softcopy Publications

LK3T-2326	CDROM	IMS/ESA Version 6 Softcopy Library
SK2T-0730	CDROM	IBM Online Library: Transaction Processing and Data
SK2T-0710	CDROM	MVS Collection
SK2T-6700	CDROM	OS/390 Collection

Index

Special Characters

\$\$IMSDIR

creating 34

updating 8, 20

/TEST MFS command 3, 13

B

backup operations, MFS library 15

batch mode (MFS Language utility) 9

C

control blocks

creation by MFS Language utility 7

MFS Language utility 3, 4

testing 13

D

DD statements

DFSWTnnn procedure 56

MFSUCT procedure 27

MFSRVC procedure 33

TCO Verification utility (DFSTVER0) 61

DELETE function (MFS Service utility DFSUTSA0) 34

DFSUDT0x (device characteristics table) 40

DFSWTnnn procedure (Spool SYSOUT Print utility
DFSUPRT0) 55

E

EXEC statement

MFS Device Characteristics Table (DFSUTB00) 26

MFS Service (DFSUTSA0) 33

Multiple Systems Verification utility (DFSUMSV0) 47

Spool SYSOUT Print utility (DFSUPRT0) 56

TCO Verification utility (DFSTVER0) 60

F

FMTCPY control statement

MFSBTCH2 procedure 12

MFSUTL procedure 9

format set 3

I

IMS.FORMAT library

backup and restore operations 15

IMS.REFERAL library

backup and restore operations 15

partitioned data set (PDS) directory 6

IMS.RESLIB library 31

IMSMSV procedure (Multiple Systems Verification utility
DFSUMSV0) 46

IMSWTnnn procedure (Spool SYSOUT Print utility
DFSUPRT0) 57

INDEX function (MFS Service utility DFSUTSA0) 34

ITB (intermediate text block) 5

L

LIST function (MFS Service utility DFSUTSA0) 37

logical links, multisystem control block 48

logical terminals, multisystem control block 45, 50

M

MFS Device Characteristics Table utility (DFSUTB00)

DD statements

DCT 27

DCTIN 27

DCTLNK 27

DEFLT5 27

PROCLIB 27

STEPLIB 27

SYSIN 27

SYSLIB 27

SYSLIN 27

SYSLMOD 27

SYSPRINT 27

SYSPUNCH 27

SYSUT1 27

description 23

EXEC statement

DCTSUF= 26

description 26

DEVCHAR= 26

DSCMSUF= 26

DSCTSUF= 26

MFS descriptor format 27

MFSUCT procedure 23

PROC statement

DCTSUF= 26

description 25

DSCMSUF= 26

DSCTSUF= 26

restrictions 23

MFS Language utility (DFSUPAA0)

batch mode

description 9

MFSBTCH1 procedure 10

MFSBTCH2 procedure 11

control blocks 3, 4

ddnames (MFSRVC)

FORMAT 21

REFIN 21

SYSIN 21

SYSPRINT 22

SYSSNAP 22

ddnames (MFSULT, MFSBTCH1, and MFSBTCH2)

DUMMY 21

FORMAT 21

MFS Language utility (DFSUPAA0) *(continued)*

- REFIN 21
- REFOUT 21
- REFRD 21
- SYSIN 21
- SYSLIB 21
- SYSUT3 21
- SYSUT4 21
- UTPRINT 21

description 3

FMTCPY control statement

- MFSBTCH2 procedure 12

- MFSUTL procedure 9

format set 3

JCL parameter descriptions

- COMPR= 19
- COMPR2= 20
- COMPR3= 20
- DEVCHAR= 20
- DIRUPDT= 20
- LN= 20
- MBR= 20
- PCOMP= 19
- PSUBS= 19
- PXREF= 19
- RGN= 20
- SN= 20
- SNODE= 20
- SOR= 20
- SOUT= 20

JCL requirements

- MFSBACK procedure 15
- MFSBTCH1 procedure 10
- MFSBTCH2 procedure 11
- MFSREST procedure 17
- MFSTEST procedure 14
- MFSUTL procedure 8

modes 3

REFCPY control statement

- MFSBTCH1 procedure 11

- MFSULT procedure 9

standard mode (MFSUTL procedure)

- phase 1 7
- phase 2 7
- preprocessor 4
- region parameter estimate 22

test mode (MFSTEST procedure)

- description 12
- phase 1 processor 13
- phase 2 14
- region parameter estimate 22
- source statement preprocessor 13

MFS Service utility (DFSUTSA0)

DD statements

- STEPLIB DD 33
- SYSRINT DD 33
- SYSSNAP DD 33

DELETE function 34

description 31

EXEC statement

- description 33

MFS Service utility (DFSUTSA0) *(continued)*

EXEC statement *(continued)*

- PARM= 33
- REGION= 33

INDEX function 34

LIST function

- output 40

LIST function output 37

MFSRVC procedure 32

PROC statement

- description 32
- DEVCHAR= 33
- SOUT= 33
- SYS2= 33

RELATE function 37

restrictions 32

SCRATCH function 35

utility control statement keywords

- DEV= 43
- DEVCHAR= 42
- DIV= 44
- FEAT= 44
- FMT= 42
- MDL= 43
- MSG= 42
- PDB= 42
- TBL= 42

utility control statement parameters

- ALL 41
- FORMAT 42
- INDEX 42
- REFER 42

utility control statements 41

MFSBACK procedure (MFS Language utility)

- backup 4

- description 15

- JCL requirements 15

MFSBTCH1 procedure (MFS Language utility)

- description 10

- JCL requirements 10

MFSBTCH2 procedure (MFS Language utility)

- description 11

- DIRUPDT= 8

- JCL requirements 11

MFSDCT 23

MFSREST procedure (MFS Language utility)

- description 15

- JCL requirements 17

- restore 4

MFSRVC procedure (MFS Service utility) 32

MFSTEST procedure (MFS Language utility)

- JCL requirements 14

- region parameter estimate 22

- step 1 (phase 1) 13

- step 1 (source statement preprocessor) 13

- step 2 (phase 2) 14

MFSUTL procedure (MFS Language utility)

- DIRUPDT= 8

- generation 3

- JCL requirements 8

- region parameter estimate 22

MFSUTL procedure (MFS Language utility) *(continued)*
 step 1 (phase 1) 7
 step 1 (preprocessor) 4, 6
 step 2 (phase 2) 7
 MSC (Multiple Systems Coupling) 45
 Multiple Systems Verification utility (DFSUMSV0)
 description 45
 EXEC statement 47
 IMSMSV procedure 46
 input validation 48
 invoking the procedure 47
 logical links 48
 logical terminals 45, 50
 MSC (Multiple Systems Coupling) 45
 multisystem control block verification 48
 multisystem path map 50
 output messages 50
 partner IDs 48
 physical links 48
 prerequisites 46
 PROC statement
 ALL= 47
 CLASS= 47
 description 47
 DSM= 47
 DSN= 47
 REG= 47
 SER= 47
 UNIT= 47
 procedure for executing 46
 processing phases 48
 restrictions 45
 SYSID paths
 description 49
 local 49
 remote 49
 transaction code attributes
 consistency between systems 49
 description 49
 local 49
 remote 49
 utility control statements
 description 52
 examples 53
 multisystem control blocks 48
 multisystem path map, MSC 50

P

partitioned data set (PDS) directory 6
 PDIR parameter (MFS Service utility) 41
 physical links, MSC 48
 PROC statement
 MFS Device Characteristics Table utility
 (DFSUTB00) 25
 Multiple Systems Verification utility (DFSUMSV0) 47
 Spool SYSOUT Print utility (DFSUPRT0) 56

R

REFCPY control statement
 MFSBTCH1 procedure 11

REFCPY control statement *(continued)*
 MFSULT procedure 9
 RELATE function (MFS Service utility DFSUTSA0) 37
 restore operations, MFS library 15
 RGN= operand PROC statement (MFS Language utility) 22

S

SCRATCH function (MFS Service utility
 DFSUTSA0) 35
 script member, error 59
 SEQBLKS data set 7
 Spool SYSOUT Print utility (DFSUPRT0)
 blocking factors 55
 DD statements
 SPOOLnn 56
 STEPLIB 56
 SYSPRINT 56
 description 55
 DFSWTnnn procedure 55
 EXEC statement 56
 IMSWTnnn procedure 57
 PROC statement
 description 56
 RGN= 56
 SOUT= 56
 SYS1= 56
 SYS2= 56
 sample output 57
 system messages 57
 SYSID paths, MSC 49

T

TCO Error Report 61
 TCO-Message-Table Report 63
 TCO script library 59
 TCO-Statistics Report 62
 TCO-Summary Report 63
 TCO Time-Schedule Request Table 62
 TCO-Timer-Elements Report 62
 testing MFS control blocks 13
 Time-controlled operations verification utility
 (DFSTVER0)
 return codes 64
 Time-Controlled Operations Verification utility
 (DFSTVER0)
 DD statements
 DFSTCF 61
 STEPCAT 61
 STEPLIB 61
 SYSIN 61
 SYSPRINT 61
 SYSUDUMP 61
 description 59
 EXEC statement 60
 output
 description 61
 error report 61
 message-table report 63
 statistics report 62

Time-Controlled Operations Verification utility
(DFSTVER0) (*continued*)
 summary report 63
 time-schedule request table 61
 timer elements report 62
TCO script library 59
TCO Verification procedure 59
transaction code attributes (MSC) 49

U

utilities

MFS Device Characteristics Table Utility
(DFSUTB00) 23
MFS Language Utility (DFSUPAA0) 3
MFS Service Utility (DFSUTSA0) 31
Multiple Systems Verification Utility
(DFSUMSV0) 45
Spool SYSOUT Print Utility (DFSUPRT0) 55
Time-Controlled Operations Verification Utility
(DFSTVER0) 59

V

verification process 48

Readers' Comments — We'd Like to Hear from You

IMS/ESA
Utilities Reference:
Transaction Manager

Publication No. SC26-8771-01

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

Company or Organization

Phone No.



Cut or Fold
Along Line

Fold and Tape

Please do not staple

Fold and Tape



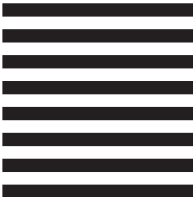
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department BWE/H3
P.O. Box 49023
San Jose, CA 95161-9945



Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold
Along Line



Program Number: 5655-158



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC26-8771-01



Spine information:



IMS/ESA

IMS/ESA V6 Utilities Ref: TM

Version 6